

**UNIVERSIDADE DO MINDELO
DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR**

CURSO DE LICENCIATURA EM INFORMÁTICA DE GESTÃO

**RELATÓRIO DE PROJETO DE LICENCIATURA
ANO LETIVO 2014/2015 – 4º ANO**

Autor: Ilucete Fernandes Tavares, N.º 1761

Mindelo, 2015

Ilucete Fernandes Tavares

PLATAFORMA ROTEIRO TURÍSTICO VIRTUAL - PRTV

APLICAÇÃO MÓVEL

Trabalho apresentado como requisito parcial
à obtenção do grau de Licenciatura em
Informática Gestão, pela Universidade do
Mindelo.

Orientador:

Eng: Samuel Santos Lima

Dedicatória

Dedico este trabalho a todos, que de um aspeto ou de outro, me ajudaram durante os anos de formação, em especial a minha mãe, os meus professores, os meus amigos e colegas.

Agradecimentos

Agradeço a minha querida mãe pelo esforço, dedicação e ajuda que sempre me deu desde do primeiro ano de formação até o último ano.

À minha irmã Crisólita Tavares do fundo do meu coração pelas ajudas e hospedagem durante todos esses anos de formação.

Ao meu professor e coordenador do curso João Dias pela ajuda, percepções e atenção para progredir com o projeto.

Ao meu namorado Ivan Matos pelo incentivo e ajuda para terminar o curso e alcançar os objetivos.

Ao meu grande amigo e colega de escola Ivan Semedo pela sua grande ajuda e atenção com o meu projeto.

Ao meu Padrinho pelo carinho e ajuda durante todo o tempo.

A minha colega de trabalho Rilda Santos por todo o apoio, carinho, empenho e disponibilidade do seu tempo a motivar-me para o término do projeto.

Aos meus professores e amigos, em particular aqueles que tiveram ação direta no adiantamento do projeto.

Ao orientador Samuel Lima pela ajuda prestada, afeição e ainda sobretudo pelas ideias que serviram para implementar o meu projeto.

Aos meus colegas de curso e amigos pela ajuda e amizade nos diversos momentos de angústia, convívio, em especial a amiga e colega Nelly Simone, e amigo Ridson Robel pela ajuda na elaboração do trabalho.

Resumo

Com o crescimento do mercado de dispositivos móveis e uma previsão de aumento significativo do turismo em São Vicente, este trabalho apresenta a conceção e o desenvolvimento de uma aplicação Android para prover e gerir roteiros turísticos a partir de uma abordagem distribuída com o armazenamento de dados na nuvem com a plataforma virtual.

O aumento dos turistas na cidade promoverá o turismo regional e a visitação aos pontos turísticos da cidade do Mindelo. A dificuldade na localização dos pontos turísticos pode ser um problema para os turistas, devido à falta de informações, sinalizações e de serviços para atender a demanda.

O uso de aplicações para dispositivos móveis vai ao encontro desta necessidade, permitindo o desenvolvimento de inúmeras aplicações turísticas. Neste contexto, este trabalho apresenta o estudo e o desenvolvimento de uma aplicação para dispositivos móveis, com sistema operativo Android, que serve como a informação virtual da cidade do Mindelo. Por meio da aplicação o turista pode se localizar no mapa, permite buscar rapidamente os pontos turísticos, ver o itinerário da linha do turismo, seus pontos de paragem e obter informações detalhadas de cada ponto turístico. Esta aplicação utiliza sistemas de localização geográfica, mapas, comunicação de dados e base de dados.

Palavras-Chave: Android, Turismo, Mapa, Serviços Baseados em Localização, Dispositivos Móveis.

Abstract

With the growth of the mobile device market and a forecast of a significant increase of tourism in São Vicente, this work presents the design and development of an android application to provide management of tourist itineraries from a distributed approach to data storage in the cloud with the virtual platform.

The increase in tourists in the city will promote regional tourism and visits to tourist sights of the city of Mindelo. The difficulty in locating this sights can be a problem for tourists, because the due to lack of information, signage and lack of professionals in the hotel industry and services to meet the demand.

The use of mobile apps meets to this need, allowing the development of numerous tourist apps. In this context, this work presents the study and development of an application for mobile devices with Android operating system, which serves as a virtual guide of the city of Mindelo. Through the tourist application can be located on the map, lets you quickly search the sights, see the itinerary of the tourist line, their staging points and detailed information of each attraction. This application uses geographic positioning systems, maps, data communication and database.

Keywords: Android, Tourism, Map, Location Based Services, Mobile Devices.

ÍNDICE

Dedicatória.....	II
Agradecimentos	III
Resumo	IV
Abstract.....	V
ÍNDICE.....	VI
Índice de Figuras	IX
Lista de Tabelas	X
Lista Acrónimos	XI
1 INTRODUÇÃO	1
1.1 Motivação	2
1.2 Apresentação do Tema.....	2
1.3 Objetivos.....	3
1.3.1 Objetivo Geral	3
1.3.2 Objetivos Específicos	3
1.4 Metodologia Utilizada	4
1.5 Estrutura do Trabalho	5
2 FUNDAMENTAÇÃO TEÓRICA	6
2.1 Sistema Operativos para Dispositivos Móveis	6
2.2 Tipos de Aplicações Móveis.....	7
2.2.1 Aplicações Nativos	8
2.2.2 Aplicação Web	8
2.2.3 Aplicações Híbridos	8
2.3 Serviços Baseados em Localização	9

2.3.1	Dispositivo Móvel (Mobile device).....	10
2.3.2	Provedor de Conteúdo (Content Provider)	10
2.3.3	Rede de Comunicação (Communication Network).....	11
2.3.4	Componente de localização (Positioning technology)	11
2.4	Aplicações Turísticos em São Vicente	12
2.5	Conceitos e Evoluções Relacionados com os Aplicativos.....	13
2.6	Engenharia de Software	13
2.6.1	Modelo Cascata	14
2.7	Informação, Tecnologias de Informação e Sistema de informação	17
2.8	Tecnologias Utilizadas.....	19
2.8.1	Android.....	19
2.8.1.1	As principais características do Android	20
2.8.1.2	Plataforma Android	22
2.8.1.3	Arquitetura do Google Android.....	23
2.8.1.4	Componentes de uma aplicação Android	27
2.8.1.5	Localização e Mapas no Android	31
2.8.2	Hyper Text Markup Language (HTML)	32
2.8.3	Unified Modeling Language (UML)	33
2.8.4	Personal Home Page (PHP).....	34
2.8.5	JavaScript	34
2.8.6	Base de Dados (BD)	35
2.8.7	Phonegap	36
2.8.7.1	Arquitetura da Aplicação PhoneGap	38
2.8.7.2	Características do Phonegap.....	39
2.8.8	Eclipse	42

2.8.8.1	Principais Caraterísticas do Eclipse.....	42
3	ANÁLISE DO SISTEMA	43
3.1	Caracterização da Plataforma Roteiro Turístico Virtual em São Vicente	43
3.2	Levantamento de Requisitos	44
3.2.1	Requisitos Funcionais (RF) dos aplicativos	45
3.2.2	Requisitos Não Funcionais (RNF).....	46
3.3	Lista de Caso de Uso	47
3.4	Modelação de dados.....	48
3.4.1	Diagrama caso de uso	48
3.4.2	Diagrama de Classes.....	49
3.4.3	Diagrama de sequência.....	50
4	PROTÓTIPO DO SISTEMA.....	54
4.1	A Plataforma Roteiro Turístico Virtual	54
4.2	Estrutura do projeto PR Turístico Virtual.....	54
4.3	Desenvolvimento de API Google <i>Maps</i>	56
4.4	Descrição das Funcionalidades do Sistema	58
4.4.1	Autenticação	58
4.4.2	Menu Principal	58
4.4.3	Interfaces Principais do Sistema.....	58
5	CONCLUSÃO	66
5.1	Dificuldades na implementação do sistema.....	67
5.2	Trabalhos Futuros	67
	REFERÊNCIAS BIBLIOGRÁFICAS	68

Índice de Figuras

Figura 1:Serviços Baseados em Localização.....	10
Figura 2: Modelo de Processo em Cascata.....	16
Figura 3: Componentes de Sistema de Informação	19
Figura 4:Arquitetura do Google Android	24
Figura 5. Ciclo de Vida de uma Activity.....	29
Figura 6: Aplicação utilizando Google Maps API no Android.....	32
Figura 7: Estrutura do PhoneGap	37
Figura 8:Arquitetura de uma Aplicação Phonegap.....	38
Figura 9:Diagrama de Caso de Uso.....	49
Figura 10:Diagrama de Classe.....	50
Figura 11:Diagrama de Sequência de Acesso ao Sistema.....	51
Figura 12: Diagrama de Sequência de Login no Sistema.....	52
Figura 13:Diagrama de Entidade e Relacionamento ao Sistema.....	53
Figura 14: Estrutura do Projeto Plataforma Roteiro Turístico Virtual	55
Figura 15:Código do mapa em API Google Maps	57
Figura 16:Localização do Mapa	57
Figura 17:Interface de Login	59
Figura 18:Interface de Cadastro de Pontos Turísticos.....	60
Figura 19:Interface do Mapa do Ponto Cadastrado	61
Figura 20:Interface de Cadastro dos Utilizadores	62
Figura 21:Interface do Mapa	63
Figura 22:Interface do Mapa Pontos de Interesses.....	64

Lista de Tabelas

Tabela 1: Quota de Mercado das Plataformas móveis	6
Tabela 2: Componentes de uma aplicação Android	27
Tabela 3: Requisitos Funcionais da aplicação	45
Tabela 4: Requisitos não Funcionais	46
Tabela 5: Lista de Caso de Uso	47

Lista Acrónimos

API - Application Programming Interface
GPS - Global Positioning System
GSM - Global System for Mobile Communication
HTML - Hyper Text Markup Language
HTC - High-Tech Computer
IDE - Integrant Development Environment
IOS - iPhone Operating System (Apple)
IRT - Inventory of Tourism Resources
JPEG - Joint Photographics Experts Group
LBS – Location Based Service
LG - Lucky Goldstar
MMS - Multimedia Messaging Service
SGBD – Data Base Management System
SDK - Software Development Kit
SMS - Short Message Service
U I - User Interface
UML - Unified Modeling Language
OHA - Open Handset Alliance
PHP - Hypertext Preprocessor

1 INTRODUÇÃO

Com o surgimento e a evolução dos dispositivos móveis, os mesmos passaram a fazer parte do dia-a-dia da vida das populações, encaminhando uma série de vantagens. Este novo cenário levou a uma mudança de paradigma relativo ao acesso á dados e serviços disponibilizados. (Fling, 2009).

O projeto foi desenvolvido para os dispositivos móveis tendo em conta que estão cada vez mais presentes na vida das populações, quer para trabalho, como para lazer, pois possuem várias funcionalidades inovadoras. Entre essas funcionalidades, se destacam o acesso a internet, GPS, acesso a redes sem fio, mapa, etc.

A aplicação desenvolvida intitulada na Plataforma Roteiro Turístico Virtual é um sistema de informação que apresenta aos turistas os locais de maior interesse na ilha, nomeadamente os restaurantes, pensões, farmácias, hotéis, discotecas, praias disponíveis, na cidade do Mindelo, com as informações úteis que possam necessitar.

A aplicação recorre ao uso de mapa para mostrar aos turistas os locais ou monumentos históricos, que se encontram na sua proximidade.

1.1 Motivação

A procura da informação turística é uma tarefa com que as pessoas lidam diariamente e cada vez mais os utilizadores pretendem ter acesso às informações com mais utilidade possível. O aumento da utilização dos dispositivos móveis em Cabo Verde, cedida a falta de meios eficientes para a apresentação de informações principalmente na área do turismo foram as principais motivações que me levaram ao desenvolvimento da plataforma roteiro turístico virtual para aplicações móveis.

1.2 Apresentação do Tema

Atualmente estão cada vez mais presentes, os dispositivos de computação móvel nos mais diversos domínios. São utilizados, enquanto ferramentas de trabalho, para acesso à entretenimento e/ou informação genérica. Por outro lado, continua a ser escassa informação de pontos de interesse turístico nas nossas cidades que possa ser acedida de forma rápida e intuitiva para um público-alvo heterogéneo.

Com as aplicações os utilizadores possuem uma ferramenta capaz de disponibilizar informações de alguns pontos turísticos na cidade do Mindelo de forma permanente permitindo que os turistas visualizem uma lista de pontos turísticos. Através desta lista o utilizador tem a possibilidade de visualizar informações com maior detalhe tendo ainda a possibilidade de atualizar os dados, inserir novos pontos turísticos e ainda visualizar uma rota, no mapa, para chegar a um determinado ponto de interesse.

Uma outra funcionalidade desta aplicação é criar um sistema que possibilita a expansão da aplicação para outras plataformas, também móveis e que permita uma integração com outros sistemas turísticos.

A criação do protótipo consiste em aprofundamento amplo de importantes áreas do conhecimento de computação, os quais serão abordados durante o decorrer deste trabalho.

1.3 Objetivos

1.3.1 Objetivo Geral

O presente trabalho tem como objetivo geral de desenvolver um protótipo de plataforma roteiro turístico virtual bem como implementar uma aplicação Android para dispositivos móveis, aceder informações de interesses turísticos, para localizar facilmente a posição atual do dispositivo e dos pontos turísticos da cidade do Minho.

A aplicação mostrará o mapa da cidade do Minho com os principais pontos turísticos como: Restaurantes, Pensões, Discotecas, Farmácias, Hotéis, Praias entre outros de forma simples, clara e objetiva. O turista ao clicar no ponto turístico receberá informações relevantes sobre o local. Também será possível realizar uma busca rápida de pontos turísticos, através de uma lista e poderá visualizar o percurso da linha turismo e seus respetivos pontos de paragem.

1.3.2 Objetivos Específicos

Pretende-se que a aplicação disponibilize aos turistas as seguintes funcionalidades:

- ❖ Criar um sistema que possibilite a expansão da aplicação para outras plataformas, também móveis e que permita uma integração com outros sistemas turísticos;
- ❖ Desenvolver o código fonte da aplicação Android com uso da linguagem de programação HTML, PHP;
- ❖ Desenvolver as interfaces de utilizadores com uso da linguagem de programação HTML;
- ❖ Realizar a conexão entre a aplicação Android e a base de dados remotamente, com o uso da linguagem PHP;

- ❖ Apresentar aos turistas alguns lugares iniciais ao chegarem á São Vicente;
- ❖ Informar aos turistas os pontos de acesso mais importantes tais como:
Hotéis, Restaurante, Praias, Discotecas, Farmácias, Pensões.
- ❖ Efetuar pesquisas de lugares (finalidade, tipo de lugar, localização, tipologia, etc.);
- ❖ Visualizar imagens dos lugares e informações detalhadas;
- ❖ Visualizar lugares;
- ❖ Contatar o gerente de cada lugar pelo envio de correio eletrónico e através de chamadas;
- ❖ Determinação automática via GPS dos monumentos e locais históricos.

1.4 Metodologia Utilizada

O processo de desenvolvimento de *software* consiste genericamente num conjunto de fases, tarefas e atividades, realizadas por intervenientes que desempenham várias funções, de modo a elaborarem diversos factos, que em conjunto contribuem para a produção de um sistema de software. A metodologia implica adicionalmente a definição de aspetos que tornam concretizável a noção de processo, designadamente a utilização de técnicas, notações e ferramentas. (SILVA e VIDEIRA, 2008).

Para o desenvolvimento deste trabalho foi utilizada a engenharia de *software* como metodologia a ser definida.

A primeira fase no desenvolvimento do projeto foi a definição do âmbito e dos objetivos, bem como o planeamento das fases do projeto e suas atividades.

De seguida o projeto teve como objetivo o conhecimento dos pontos turísticos para que os turistas localizam no mapa.

Depois foram realizadas algumas buscas dos pontos turísticos para escolher os lugares interessados na implementação do sistema *Android*.

Depois de conhecer os pontos turísticos foi elaborada uma lista de requisitos necessários para o funcionamento do sistema.

A quarta fase consiste na modelação dos dados para assegurar a completa e correta definição de funcionalidades e a satisfação das necessidades dos turistas.

A última fase do projeto consiste no desenvolvimento do protótipo do sistema.

O desenvolvimento da aplicação Android requer a utilização de diversas tecnologias, descritas nas secções seguintes. Para cada uma delas é apresentado um pequeno exemplo de utilização, com código escrito na linguagem de programação HTML.

1.5 Estrutura do Trabalho

De entre os objetivos definidos, o trabalho encontra-se dividido em quatro capítulos.

Antes de iniciar os capítulos temos a introdução, a motivação, apresentação do tema, objetivos gerais, objetivos específicos e a metodologia utilizada.

No primeiro capítulo apresenta a fundamentação teórica com os sistemas operativos para dispositivos móveis, os serviços baseados em localização, as aplicações turísticos em são vicente, sistema de informação entre outros pontos importantes do trabalho.

No segundo capítulo fez-se a abordagem da análise do sistema, análise dos requisitos e a modelagem do sistema através dos diagramas.

No terceiro capítulo fez-se o desenvolvimento do protótipo e a demonstração de alguns códigos do Google maps, algumas interfaces do protótipo abordado neste trabalho referente a interação do turista com o sistema.

No quarto capítulo apresenta-se a conclusão geral do trabalho, que descreve as considerações finais e sugestões de implementações futuras.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Sistema Operativos para Dispositivos Móveis

O mercado de *Smartphones* em todo o mundo cresceu 13,0% ano a ano em 2015 Q2, com 341,5 milhões de embarques, de acordo com dados do *International Data Corporation* (IDC). Este crescimento deve-se principalmente aos ganhos de experiência em mercados emergentes, como APEJ e MEA.

Os embarques deste trimestre foram ligeiramente inferiores às previsões e a IDC espera ver uma redução perceptível nos embarques de *Smartphones* em 2015 em que a China se junta a América do Norte e Europa Ocidental, num padrão de crescimento mais maduro. Android domina o mercado com uma quota de 82,8% em 2015 Q2. Samsung reafirmou sua liderança global com um foco renovado em *Smartphones* de baixo custo.

A tabela seguinte mostra os sistemas operativos para dispositivos móveis.

Período	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Tabela 1: Quota de Mercado das Plataformas móveis

Fonte: [idc]

Android - dominou o mercado de *Smartphones* com uma quota de 82,8%. O Samsung, teve volumes inferiores ao trimestre anterior e ao período homólogo. Isto vem no meio de um desempenho abaixo do esperado por seus lançamentos emblemáticas, S6 e S6 *Galaxy*

Borda. No entanto, a participação Android tem visto um aumento em relação a 2015Q1, com um forte crescimento em vendas unitárias por outros operadores, como a *Huawei*, *ZTE* e *Xiaomi*.

iOS - assistiu a sua quota de mercado para 2015Q2 com uma descida de 22,3% no trimestre com 47,5 milhões de embarques. Apesar da descida sazonal, a Apple obteve sucessos graças ao apetite insaciável dos consumidores para os dispositivos *iOS*. A popularidade do *iPhone 6 Plus* continuou em muitos mercados-chave, incluindo a China, onde o mercado global de *Smartphones* viu um crescimento de 6,7%.

Windows Phone - teve uma descida no trimestre de 4,2%, com um total de 8,8 milhões de unidades vendidas neste trimestre. Desde a aquisição da Nokia em 2014, a Microsoft vem reformulando o portfólio de produtos com dispositivos de marca Lumia Microsoft.

A marca Acer é uma nova entrada no top cinco neste segmento. A maioria dos outros fornecedores perdeu muito nos embarques no trimestre, com exceção da Samsung, que mostrou um aumento de 8,5% com a sua gama ATIV de telefones.

BlackBerry OS - teve um pequeno aumento em algumas regiões, mas continuou a diminuir seu crescimento ao nível mundial. A maior parte das suas vendas em volume veio do *Blackberry Classic*.

2.2 Tipos de Aplicações Móveis

Segundo CHEDE (2013), existem três tipos de aplicações (apps) móveis, com as suas potencialidades, restrições e cuidados especiais:

- ❖ Aplicações Nativos;
- ❖ Aplicações Web;
- ❖ Aplicações Híbridos.

2.2.1 Aplicações Nativos

Uma app nativo é uma plataforma ou conjunto de dispositivos específicos e desenvolvido pelo modelo de programação desta plataforma.

O desenvolvimento de uma aplicação nativo, ou seja, focado em uma plataforma específica, significa que podemos explorar ao máximo a potencialidade da plataforma e dos seus dispositivos.

Esta aplicação é obtida de um marketplace ou app store como exemplo a App Store da Apple ou Google Play para Android. Como a aplicação tem acesso direto aos recursos da plataforma: câmeras, compasso, acelerômetro, calendário e lista de contatos etc., consegue-se tirar o máximo dos aparelhos. Também utiliza os componentes básicos de UI (User Interface) da plataforma, como botões e ícones, bem como suas interfaces gestuais.

2.2.2 Aplicação Web

As aplicações web são acedidas através de browsers, como muitas das aplicações web do desktop (ambiente de trabalho).

Mas essa aplicação tem de conhecer o ambiente em que ela está operando e ajustar-se de forma adequada. A interface que aparece em um desktop deve ser diferente do que aparecer no tablet.

Entre os aspetos positivos temos a facilidade de desenvolvimento, uma vez que não demanda códigos únicos por plataforma, dado que são inerentemente cross-platform (reduz custos, naturalmente), e como é acedido via URL pode ser integrado em sistemas web como Twitter e Facebook.

2.2.3 Aplicações Híbridos

As aplicações híbridas contêm dois elementos: uma componente web, baseado em HTML5, e um container ou “bridge” nativo, que permite aceder os recursos essenciais da

plataforma e dispositivos.

O container constitui em um conjunto de APIs que permite o app HTML5 aceder os recursos nativos do smartphone ou tablet. A app híbrida tem seu código principal desenvolvido em HTML5.

Uma vantagem significativa é que as entidades podem utilizar o HTML5 para diminuir os custos de desenvolvimento cross-platform, deixando apenas parte do código que explora a funcionalidade específica da plataforma ser escrita no modelo de programação da própria plataforma.

2.3 Serviços Baseados em Localização

A definição típica de serviços baseados em localização são para serviços de informação, acessíveis por dispositivos móveis através das redes de dados fazendo uso da posição geográfica do dispositivo.

Atualmente a interação do utilizador e a possibilidade de gerar conteúdo são fundamentais para os serviços e aplicações. Uma melhor definição atual de serviço baseado em localização é um serviço que o utilizador é capaz de determinar sua localização, as informações fornecidas são especialmente relacionadas com a localização do utilizador, é oferecido ao utilizador uma interação dinâmica ou bidirecional com o provedor de conteúdo.

São quatro os componentes comuns a todas as aplicações *Location Based Services* (LBS): dispositivo móvel, provedor de conteúdo, rede de dados e um componente de localização, conforme apresenta na Figura1.



Figura 1: Serviços Baseados em Localização

Fonte: [Ferraro e Aktihanoglu]

2.3.1 Dispositivo Móvel (Mobile device)

Os *Smartphones* tiveram uma importância na rápida absorção de serviços LBS, dado seu tamanho da interface, geralmente grandes e a inclusão quase que universal de tecnologias de localização (como GPS). O rápido crescimento dos *netbooks* e dos *tablets* também fizeram os desenvolvedores começarem a pensar em dispositivos móveis muito mais do que apenas telefones celulares.

2.3.2 Provedor de Conteúdo (Content Provider)

Um provedor de conteúdo é uma entidade que cria ou detém o conteúdo que pode ser fornecido aos dispositivos móveis, diretamente ou através de terceiros.

Um provedor LBS não armazena ou mantém todo o conteúdo e dados que são acedidos pelo utilizador no dispositivo móvel. Um exemplo são os dados de mapas, que

normalmente são fornecidos por um provedor de mapas. Cada vez mais os dados são disponibilizados aos utilizadores na forma de camadas de mapas, através de provedores de terceiros, que normalmente podem ser ligados ou desligados pelo utilizador.

2.3.3 Rede de Comunicação (Communication Network)

O desenvolvedor da aplicação LBS não tem controlo direto sobre a rede de comunicação, mas pode gerir o tráfego dos dados utilizados pela aplicação, maximizando a velocidade de transferência ou minimizando a dissimulação, bem como limitando as taxas de dados para clientes pré-pagos.

2.3.4 Componente de localização (Positioning technology)

O componente de localização refere-se à tecnologia que permite a localização do aparelho e tem a capacidade de repassar essa informação para a aplicação. Este componente está se tornando cada vez mais escondido, até mesmo para os desenvolvedores das aplicações, mas ainda há um grau de escolha entre os métodos de localização, que podem ser por triangulação, Cell ID, navegação por satélite ou por *Wireless Positioning System* (WPS).

Nos dispositivos que possuem mais de uma tecnologia de localização disponíveis, a utilização híbrida, de posicionamento, vem se tornando cada vez mais utilizada para minimizar as desvantagens do hábito de uma única tecnologia.

Para o desenvolvimento do protótipo foi necessário adquirir o conhecimento sobre *Android*, que abrange todas as informações básicas e necessárias sobre o sistema operativo móvel *Android*, sistema de informação para aplicativos móveis, engenharia de *software* e definir as ferramentas necessários para sua estrutura.

2.4 Aplicações Turísticas em São Vicente

Neste contexto será abordado o estudo de algumas aplicações para dispositivos móveis, voltados para o setor turístico em São Vicente, que já foram desenvolvidos até o presente momento e que também tem como objetivo auxiliar o turista na localização dos pontos de interesse numa determinada região.

Quando se pensa em turismo, normalmente o que primeiro se lembra é de hotéis, restaurantes, praias e pouco mais. No entanto, o turismo engloba muito mais do que se possa imaginar á primeira vista. Com efeito, tudo o que seja capaz de motivar a deslocação de pessoas, ocupar os seus tempos livres ou satisfazer às necessidades da sua permanência em um local pode ser entendido como recurso turístico.

A inventariação dos recursos com interesse para o turismo, servirá como ponto de partida para a criação de produtos turísticos locais, no sentido de maximizar as potencialidades de cada município. Para desenvolver as potencialidades turísticas de um município é imprescindível que haja informações confiáveis e de qualidade, que permitirão análises e decisões acertadas.

Assim, o Inventário dos Recursos Turísticos (IRT) representa um instrumento valioso para o planeamento turístico, uma vez que servirá de base para a elaboração de estratégias, planos e programas adequados à realidade e necessidades do município.

O IRT de São Vicente deverá constituir um reflexo fiel da realidade dos recursos turísticos existentes, indicando a informação técnica e a situação em que se encontram, sendo que através deste instrumento será possível conhecer a real magnitude do património turístico do Concelho de São Vicente.

Com o objetivo de perspetivar o desenvolvimento sustentado do turismo, a Direção Geral do Turismo propôs-se a fazer o Inventário de Recursos Turísticos do Concelho de São Vicente, instrumento que constitui no registo de todos os elementos turísticos que pela sua qualidade natural, cultural e humana podem ter interesse para a estruturação da oferta turística local, pelo que representa um instrumento valioso para o planeamento turístico, uma vez que serve como ponto de partida para realizar estudos e estabelecer prioridades, necessárias para a criação dos produtos turísticos locais.

2.5 Conceitos e Evoluções Relacionados com os Aplicativos

Neste contexto serão apresentados alguns conceitos fundamentais relacionados ao trabalho. Principalmente o estudo das tecnologias empregadas no desenvolvimento das aplicações móveis, que utilizam serviços baseados em localização, mostrando em que áreas e como pode ser utilizado. Também serão expostos os conceitos de mobilidade e dispositivos móveis, trazendo dados da participação de diferentes plataformas no mercado.

2.6 Engenharia de Software

Segundo o Autor Roger S. Pressman, o *software* é um sistema lógico que contém uma sequência de instruções (programas de computadores) que ao serem executadas produzem uma função desejada.

A engenharia de *software* é a criação e a utilização de sólidos princípios de engenharia a fim de obter um software de maneira económica, que seja confiável e que trabalhe eficientemente em máquinas reais.

A engenharia de *software* consiste na utilização de um conjunto processos, métodos e técnicas de forma sistemática, disciplinada, e quantificado ao desenvolvimento, operação e manutenção de um sistema.

As atividades associadas a Engenharia de *Software* podem ser agrupadas em três grandes fases, tendo em conta que o seu objetivo é o desenvolvimento e operação de um produto: conceção, implementação e manutenção. Cada uma destas fases pode ainda ser dividida em outras mais elementares. Ao longo de cada fase existem tarefas, subprodutos a desenvolver, pontos de verificação e intervenientes.

Existe também um conjunto de atividades de suporte contínuo: gestão de projeto, controlo de qualidade, gestão da configuração, elaboração de documentação, elaboração de estimativas, gestão de riscos entre outras. A engenharia de *software* é uma área de conhecimento muito vasto, o que torna, ainda mais difícil, a sua aplicação de forma rigorosa e sistemática.

Processos, métodos e ferramentas

Engenharia de *software* é uma tecnologia em camadas. A gestão de qualidade total e filosofias análogas levam á cultura de um processo contínuo de aperfeiçoamento e essa cultura, em última análise, leva ao desenvolvimento de abordagens cada vez mais amadurecidas, para a engenharia do *software*.

O fundamento da engenharia de *software* é a camada de processo. O processo de engenharia de *software* é o adesivo que mantém unidas as camadas de tecnologia e permite o desenvolvimento racional e oportuno do software para computador. O processo define uma estrutura para um conjunto de áreas chave de processo (key process áreas, KPA) que deve ser estabelecido, para a efetiva utilização da tecnologia de engenharia de *software*.

2.6.1 Modelo Cascata

O Modelo Clássico ou Cascata - é um modelo de desenvolvimento de *software* sequencial, no qual o desenvolvimento é visto como um fluir constante para frente (como uma cascata) através das fases de análise de requisitos, projetos, implementação, testes,

integração e manutenção de software. A origem do termo *cascata* é frequentemente citada como sendo um artigo publicado em 1970 por Royce, que defendia uma abordagem iterativa para o desenvolvimento de *software*.

O modelo cascata é um dos mais importantes modelos e é uma referência para muitos outros, servindo de base para muitos projetos modernos.

A grande parte do sucesso do modelo cascata está no facto dele ser orientado para documentação. No entanto salienta-se que a documentação abrange mais do que o arquivo de texto, ou seja abrange representações gráficas ou mesmo simulação.

O modelo Cascata aplica-se bem em situações em que o *software* a ser desenvolvido é simples, os requisitos são bem conhecidos, a tecnologia utilizada é bem acessível e os recursos para o desenvolvimento, estão disponíveis.

O modelo em cascata segue cinco etapas para o desenvolvimento do *software*, como mostra a figura 2:

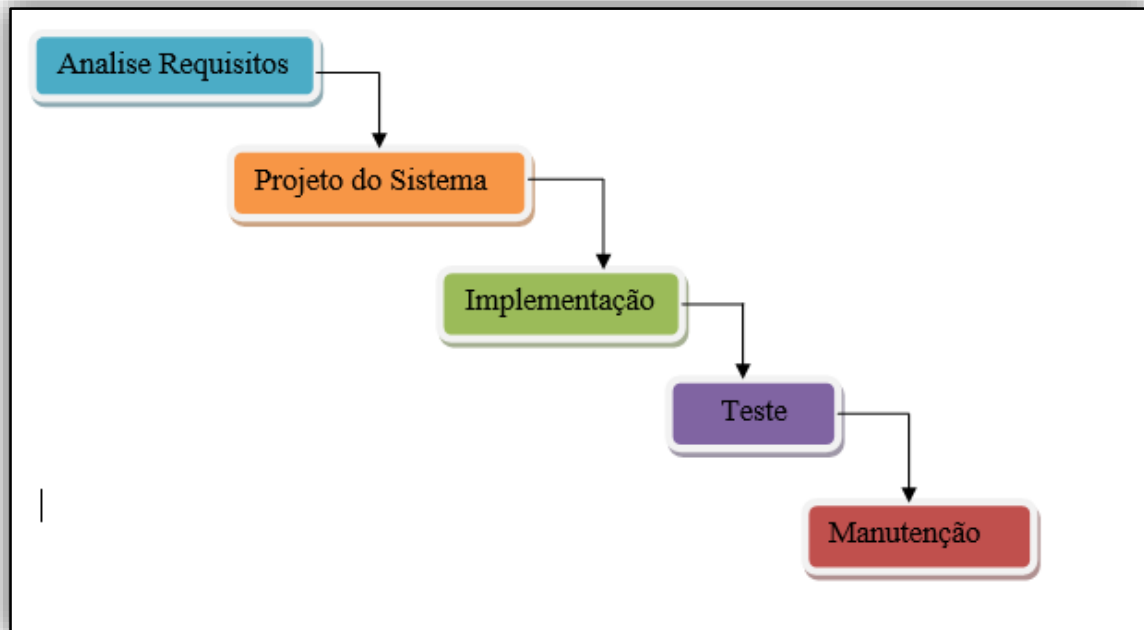


Figura 2: Modelo de Processo em Cascata

- **Análise dos Requisitos** - nesta sequência estabelecem-se os requisitos dos produtos que se deseja desenvolver, o que consiste usualmente nos serviços que se devem fornecer, limitações e objetivos do *software*.
- **Projeto do sistema** - é um processo de vários passos, que se centraliza em quatro atributos diferentes do sistema: estrutura de dados, arquitetura do *software*, detalhes de procedimentos e caracterização das interfaces. O processo de projeto representa os requisitos, de uma forma, que permita a codificação do produto.
- **Implementação** - esta é a fase em que se produz efetivamente o código, na forma de um sistema de *software* funcional. Pode sugerir a inclusão de um teste unitário dos módulos nesta etapa, nesse caso, as unidades de código produzidas são testadas individualmente, antes de passar para a próxima etapa.

- **Teste do sistema** - processo de codificação, fase de teste do sistema. O processo de teste centraliza-se em dois pontos principais: as lógicas internas do *software* e as funcionalidades externa.
- **Manutenção** - Esta etapa consiste na correção de erros, que não foram previamente detetados, em melhorias funcionais.

Todas as modificações do modelo cascata possuem os mesmos conceitos básicos: a ideia que uma etapa fornece saídas que serão utilizadas como entradas, para a etapa seguinte, portanto, o processo de desenvolvimento de um produto de software, de acordo com o modelo Cascata, é simples de conhecer e controlar.

Outras atividades que também são levadas em consideração em cada uma das etapas de desenvolvimento do *software*: a documentação, a verificação e a administração das etapas serem documentos. A verificação, por sua vez, é necessária para que uma etapa forneça os dados corretos para a etapa seguinte.

2.7 Informação, Tecnologias de Informação e Sistema de informação

Segundo os Autores **Luiz Amaral** e **João Varajão** a Informação, Tecnologias de Informação e Sistemas de Informação, apesar de serem termos banalizados na linguagem comum, são conceitos sem um entendimento universal, pelo que se julga pertinente apresentar aqui definições que sejam simultaneamente rigorosas e próximas do que é geralmente aceite.

Também sendo mais pragmáticas, logo de maior utilidade, não comprometem, contudo, o enquadramento teórico até aqui adotado.

A **Informação** é um conjunto de dados, que quando fornecidos de forma e a tempo adequado, melhoram o conhecimento da pessoa que os recebe, ficando mais habilitada a desenvolver determinada atividade ou a tomar determinada decisão.

Sistema de Informação é um sistema que reúne, guarda, processa e faculta informação relevante para as organizações, de modo que a informação é acessível e útil para aqueles que a querem utilizar, incluindo gestores, funcionários e clientes.

Um sistema de informação é um sistema de atividade humana social que pode envolver ou não a utilização de computadores.

Ainda que conceptualmente seja aceitável a existência de um sistema de informação sem a participação de computadores, a observação da realidade permite concluir que são muito raras as organizações que não interagem computadores no seu sistema de informação. Aceitando a presença das tecnologias de informação como participantes nos sistemas de informação, podem-se redefinir, como uma perspetiva mais organizacional, segundo esses autores.

Sistema de Informação é uma combinação de procedimentos, informação, pessoas e tecnologia de informação organizadas, para o alcance de objetivos de uma organização.

As pessoas realimentam o SI com os dados que geram informações;

Interagem diretamente com o SI e utilizam as informações geradas na tomada de decisão.

As organizações são unidades que exercem diferentes funções, tais como vendas, produção, formação, etc.

As tecnologias de informação e comunicação são recursos, através dos quais os dados são transformados em informação que pode ser *hardware*, software e comunicações.

Segundo Stair e Reynolds (2006), um sistema de informação é um conjunto de componentes inter-relacionados que recolhem, manipulam e disseminam dados e informações para proporcionar um mecanismo de realimentação para atingir um objetivo. A figura 3 mostra os componentes de um sistema de informação.



Figura 3: Componentes de Sistema de Informação

Fonte: [Adaptado de Oliveira]

2.8 Tecnologias Utilizadas

Segundo o Autor Alberto Silva (2001) para o desenvolvimento do protótipo foram utilizadas várias tecnologias, como o Android, Google Maps, localização de mapa no Android, as linguagens de programação HTML, PHP, JAVA SCRIPT e banco de dados MySQL para o armazenamento de dados, UML para modelação de dados e Phonegap para desenvolver o código do sistema.

2.8.1 Android

Android é um sistema operacional de código aberto para dispositivos móveis e utiliza uma versão modificada do Sistema Operacional Linux.

Atualmente, é desenvolvida pela *Open Handset Alliance*, liderada pela Google e outras empresas.

Permite também aos desenvolvedores criarem aplicações, Java que controlam o dispositivo através de bibliotecas desenvolvidas pelo Android. É projetado principalmente para dispositivos móveis, com ecrã sensível ao toque, como *Smartphones* e *tablet*.

Em Novembro de 2007, a *Open Handset Alliance* foi apresentada ao mercado internacional como uma aliança de empresas com o objetivo de desenvolver padrões abertos para dispositivos móveis.

Apesar de ser recente (seu lançamento foi em 2008), o Android foi adotado rapidamente por diversos fabricantes de dispositivos móveis e atualmente é a plataforma que mais cresce no mundo.

2.8.1.1 As principais características do Android

O Google disponibiliza o código fonte do *Android* de forma aberta e gratuito, baseado numa licença Apache. Isto significa que qualquer pessoa, que queira utilizar o *Android*, pode simplesmente descarregá-lo da web. Os fabricantes de *hardware* (por exemplo, Samsung, HTC, Sony Ericsson, LG, etc.) podem também utilizar o SO de forma gratuita nos seus produtos e personalizar livremente o seu código.

❖ Interface

A interface padrão do utilizador no Android é baseada na manipulação direta. Essa manipulação é desenhada para ser imediata e produzir uma sensação de fluidez, utilizando constantemente a resposta háptica para informar o utilizador sobre a conclusão do comando.

❖ Aplicações

As aplicações estendem funcionalidades dos dispositivos que são desenvolvidos na linguagem Java, utilizando o sistema de desenvolvimento do *software* Android SDK que inclui um conjunto de ferramentas de desenvolvimento como o emulador, biblioteca ou computação baseados em documentação, códigos de exemplos e tutoriais. O ambiente de desenvolvimento integrado ou IDE oficialmente suportado é o Eclipse utilizando o *plugin* do Android (ADT).

O Android tem crescido na quantidade de aplicações de terceiros desenvolvidos para o sistema.

❖ Gerenciar Memória

Os dispositivos Android são normalmente movidos à bateria, o sistema gera a quantidade de memória (RAM) com o intuito de deixar a quantidade de energia gasta no mínimo, em confronto com o desktop que normalmente os sistemas assumem estarem conectados a fontes ilimitadas de eletricidade domestica.

Quando a aplicação não está sendo utilizada, o sistema automaticamente o suspende da memória, enquanto a aplicação está tecnicamente "aberta", e as aplicações suspensas não consomem qualquer recurso do sistema. Isto traz um benefício duplo, que pode aumentar a responsividade do sistema.

Outra característica do Android é aceder milhares de aplicações. Através da aplicação *Play Store*, que vem pré-instalada nos dispositivos Android, os utilizadores pode descarregar aplicações gratuitos ou pagos para o seu dispositivo, de lojas de aplicações *online* como a Google Play.

A Google Play é a loja mantida pela Google para distribuição de aplicações, jogos, filmes, música e livros.

Em termos de *hardware/software*, os dispositivos Android possuem as seguintes

características:

- ❖ **Armazenamento** - suporte SQLite (base e dados relacional);
- ❖ **Conectividade** - suporte para Bluetooth, WiFi, etc.
- ❖ **Mensagens** - suporte de SMS e MMS;
- ❖ **Navegação Web** - baseado no motor de renderização de páginas *Web Webkit* e ainda suporte para *flash* (Android 4.3);
- ❖ **Multimédia** - suporte para vários formatos de imagem (JPEG, PNG), áudio (MP3) e vídeo (MPEG4);
- ❖ **Hardware** - câmaras de vídeo, ecrã sensível ao toque, acelerómetros, GPS, gráficos 3D;
- ❖ **SDK** - contem as ferramentas necessárias para criar, compilar, testar, e distribuir aplicações Android.

2.8.1.2 Plataforma Android

Nesta parte será apresentada uma visão geral da plataforma Android, na qual a aplicação deste trabalho será implementada.

Atualmente há 1,5 bilhões de televisões em todo o mundo, um bilhão de pessoas com acesso a Internet e quase 3 bilhões de pessoas que possuem um telefone celular, isso

corresponde a mais ou menos metade da população mundial, sendo considerado um dos mais bem-sucedidos produtos de consumo (OHA, 2013).

Android é uma nova plataforma de desenvolvimento para aplicações móveis, como *smartphones* baseados no sistema operacional *Linux* que possui uma interface visual rica, GPS, diversas aplicações pré-instaladas e um ambiente de desenvolvimento bastante poderoso e que utiliza a linguagem de programação Java.

O *Android* foi desenvolvido pela *Open Handset Alliance* (OHA), liderada pelo Google e empresas como *HTC*, *LG*, *Motorola*, *Samsung*, *Sony Ericsson*, *Toshiba* e muitas outras, que tiveram por objetivo padronizar uma plataforma de código aberto e livre para celulares, para atender as necessidades do mercado atual (LECHETTA, 2010).

Atualmente, a plataforma Android é mantida pela OHA (*Open Handset Alliance*), um grupo formado por mais de 40 empresas que se uniram para inovar e acelerar o desenvolvimento de aplicações e serviços para dispositivos móveis, trazendo aos consumidores uma experiência mais rica em termos de recursos e menos onerosa financeiramente para o mercado. Pode-se dizer que a plataforma Android é a primeira plataforma completa, aberta e livre para dispositivos móveis.

O *Android SDK* é o kit de desenvolvimento que disponibiliza as ferramentas necessárias para desenvolver aplicações de plataforma Android, utilizando a linguagem Java.

O *Android* é o principal sistema operacional para dispositivos móveis, seguido pelo *iOS* da *Apple*, *BlackBerry OS*, *Windows Phone*, conforme estudo de mercado produzido pela empresa *Canalys* (CANALYS, 2013).

2.8.1.3 Arquitetura do Google Android

O sistema operacional Android é visto como uma pilha de *softwares*.

A sua arquitetura é composta basicamente por cinco camadas, onde cada camada agrupa vários programas, que suportam funções específicas do sistema, das quais:

- ❖ *Applications;*
- ❖ *Application Framework;*
- ❖ *Libraries;*
- ❖ *Android Runtime;*
- ❖ *Linux Kernel (RABELLO, 2009).*

A Figura 4 representa cada uma dessas camadas.

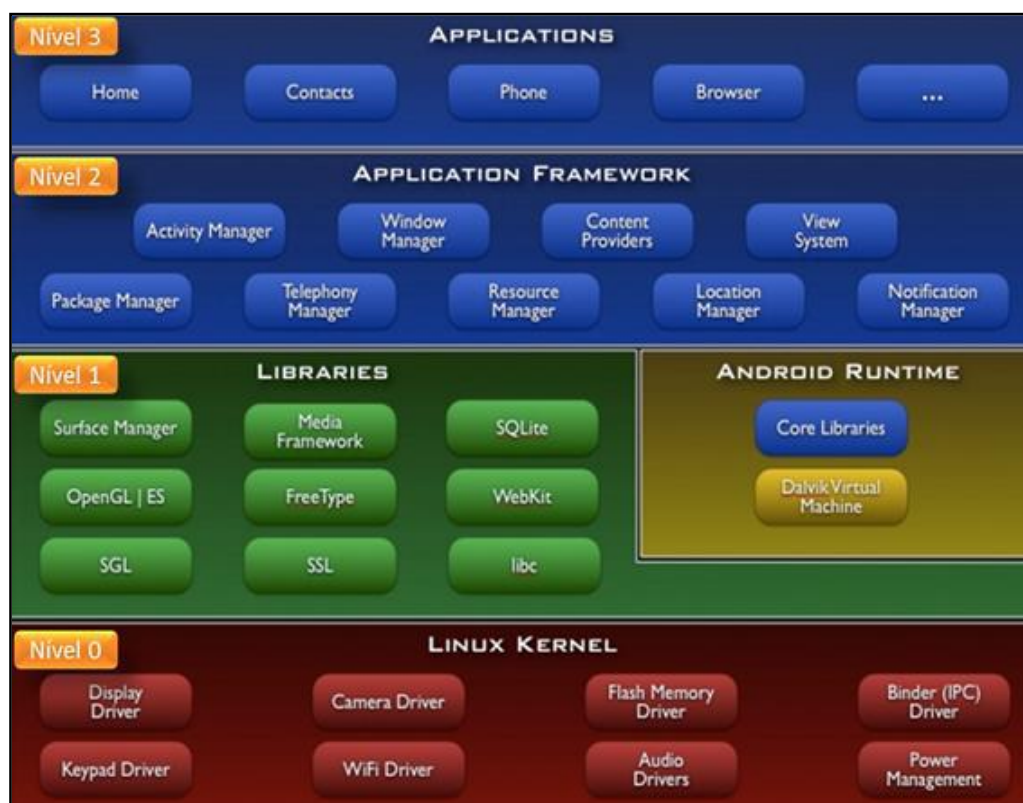


Figura 4:Arquitetura do Google Android

Fonte: Rabello (2009)

A camada **Applications** - é a camada mais alta contém as aplicações que vêm integradas nos dispositivos Android (por exemplo, *Phone*, *Contacts*, *Browser*, entre outros.). Todas as aplicações escritas neste trabalho irão ser integradas nesta camada.

Application Framework - expõe as várias capacidades do Sistema Operativo Android

para os programadores possam fazer uso delas nas suas aplicações.

Fazem parte dessa camada os seguintes componentes:

- ❖ Conjunto de componentes gráficos que pode ser reutilizados para construir uma aplicação, bem como listas, *grids*, caixas de texto, botões e até um navegador *web*;
- ❖ Provedores de conteúdo que provêm às aplicações acederem dados de outras aplicações ou partilhar seus próprios conteúdos;
- ❖ Gerir os recursos que prove acesso a recursos não-codificados como *strings*, gráficos, e arquivos de *layout*;
- ❖ Gerir notificações que permitem todas as aplicações exiba mensagens de alerta personalizáveis na barra de *status*.

A camada seguinte é subdivida no grupo das **bibliotecas (*libraries*)** e o ambiente de execução (***android runtime***) da plataforma Android, composto pelas bibliotecas padrão e pela máquina virtual denominada *Dalvik*. No primeiro grupo estão as bibliotecas escritas em C/C++, que são compostas por uma coleção de bibliotecas, que são utilizadas pela plataforma Android.

Estas bibliotecas são:

- ❖ **Biblioteca de sistema C** - é uma implementação da biblioteca C padrão (*libc*), otimizada para dispositivos que suportam a plataforma Linux (*embedded-linux*);
- ❖ **Bibliotecas de Mídias** - as bibliotecas suportam execução e gravação da maioria dos formatos de áudio e vídeo, bem como exibição de imagens.

- ❖ **Gerir Superfície** - é responsável pelo acesso ao display do dispositivo e camadas de gráficos 2D e 3D de múltiplas aplicações;
- ❖ **SGL** - é uma *engine* de gráficos 2D;
- ❖ **3D libraries** - é uma implementação baseada na especificação Open GL ES 1.0, a qual utiliza tanto a aceleração de *hardware* 3D e um avançado e otimizado *software* para o processo final dos modelos tridimensionais;
- ❖ **SQLite** - é uma biblioteca forte e leve *engine* de base de dado relacional disponível para todas as aplicações.

No que diz respeito ao ambiente de execução, a plataforma é composta pela máquina virtual *Dalvik*. Toda e qualquer aplicação em Android rodam dentro de seu próprio processo, isto é, no contexto da sua instância de máquina virtual (MV). Esta VM foi escrita para que os dispositivos possam suportar múltiplas máquinas virtuais eficientemente.

- ❖ **Linux Kernel** - baseado no *kernel* do *Linux* 3.x (Android 4.0 e superior) é o nível mais baixo do SO e contém os drivers para os componentes de hardware existentes num dispositivo Android.

2.8.1.4 Componentes de uma aplicação Android

As aplicações Android são compostas por um ou mais componentes de aplicação.

A tabela representa a lista dos quatro possíveis componentes de uma aplicação Android.

Funcionalidade	Classe Base Java	Exemplos
Foco em coisas que o utilizador possa fazer	Activity	Editar uma nota, jogando um jogo.
Execução de processos em background	Service	Tocar uma música.
Receção de mensagens	BroadcastReceiver	Desencadear alarme após evento.
Armazenamentos/recuperação de dados	ContentProvider	Abrir um contacto

Tabela 2: Componentes de uma aplicação Android

- **Activity**

As **Activities** representam ecrãs de uma aplicação com uma interface gráfica de utilizador. *Activities* funcionam como mediadores que definem como as informações serão apresentadas ao utilizador, além de controlar o fluxo da aplicação. Elas podem interagir com o utilizador e trocar informações com outras *activities* ou *services* (MEIER,2009).

A maioria do código que escrevemos para uma aplicação Android irá executar no contexto de uma *activity*. Os *Activities* normalmente correspondem os ecrãs: cada *activity* mostra uma interface para o utilizador. Quando esta não está em execução, o sistema operacional pode eliminá-la para liberar memória.

Ciclo de vida de uma *Activity*

Todo o componente do sistema que interage de qualquer modo com o utilizador deverá ter pelo menos uma ou mais *activities*. No Android apenas uma *Activity* pode estar ativa, por isso existe a *Activity Stack*, ou Pilha de Atividade para efetuar o controlo de todas as *activities*. As *activities* vão fazendo parte da pilha à medida que vão sendo executados.

Quando uma aplicação se inicia, a *Activity* principal entra em execução e é colocada no topo da pilha. Ao iniciar uma nova *Activity*, a *Activity* principal perde foco e fica em baixo da *Activity* nova que é colocado no topo da pilha.

Sendo assim sempre que uma nova *Activity* inicia, é colocada no topo da pilha e a *Activity* que estava ativa fica logo abaixo da *Activity* nova.

O ciclo de vida de uma *Activity* compreende em um conjunto de ações que são tomadas quando uma ou mais aplicações estão em execução. A figura 5 ilustra o fluxograma do ciclo de vida de uma *Activity*.

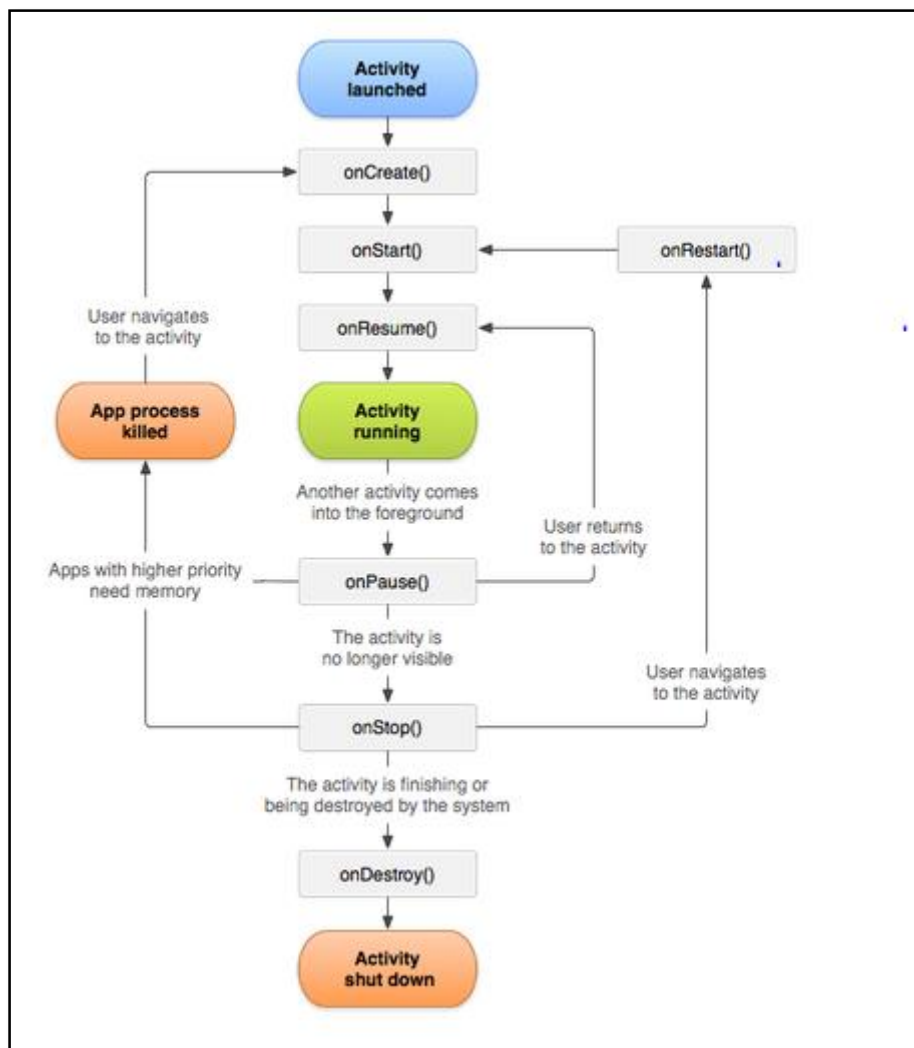


Figura 5. Ciclo de Vida de uma Activity

Em geral, o ciclo de vida de uma Activity compreende entre os seguintes métodos:

- ❖ **OnCreate()** – é o primeiro método a ser executado quando uma *Activity* se inicia pela primeira vez, é de uso obrigatório e invocado apenas uma única vez. Este método é também utilizado para carregar os *layouts* XML, carregar os dados a serem exibidos, inicializar variáveis;

- ❖ **OnStart()** - é chamado quando a *activity* está ficando visível e já tem uma interface definida. Este método indica que a Activity está prestes a ser exibida na interface;
- ❖ **OnResume()** - é chamado quando a *activity* foi parada temporariamente e está retornando à execução. É chamado após o método OnStart().
- ❖ **OnPause()** - é chamado quando a *activity* está sendo tirada do topo da execução. Geralmente é utilizado para salvar o estado da aplicação. Neste método a Activity atual é suspensa, ficando à espera de uma futura retomada para o topo da pilha;
- ❖ **OnStop()** - é chamado quando a *activity* não está mais visível e está em segundo plano. A Activity em causa está sendo destruída porque outra Activity, seja existente ou nova, está indo para o primeiro plano;
- ❖ **OnDestroy()** - executa os últimos processamentos antes do *activity* ser literalmente encerrada.

- **Service**

Os *services* - são executados em segunda plano para operações de longa duração ou controladas por processos remotos e tipicamente não fornecem uma interface gráfica. Como exemplo é um serviço de autenticação/ autorização a uma aplicação.

- **Broadcast Receiver**

O **Receiver ou broadcast** - é um componente que funcionam como um recetor de ocorrências de eventos, que podem ser próprios do sistema ou das aplicações, reage transmitindo informações e responde a transmissão de mensagens do sistema. Existem várias transmissões provenientes do sistema, por exemplo uma transmissão anunciada

que a bateria está fraca ou que uma determinada imagem foi capturada.

- ***Content Provider***

O ***Content Provider*** - é o componente utilizado pela plataforma Android para aceder dados de um aplicativo Android, para partilhar dados com outras aplicações que se encontram em execução no dispositivo, ou seja, gerem um conjunto de dados na aplicação.

Os dados podem ser armazenados no sistema de ficheiros, numa base de dados *SQLite*, na web, ou em qualquer outro local de armazenamento persistente a que a aplicação possa aceder.

2.8.1.5 Localização e Mapas no Android

Uma das funcionalidades do Android que chamam atenção é a integração com o Google Maps e a possibilidade de desenvolver aplicações de localização com GPS, com poucas linhas de código.

O componente principal do *framework* de localização é o *LocationManager system service*, que providencia as APIs para determinar a localização e a direção do dispositivo. Para utilizar é necessário solicitar uma instância, para o sistema chamando o *getSystemService (Context.LOCATION_SERVICE)* que retorna um *handle* para uma nova instância do *LocationManager*.

Com as APIs do Google Maps para Android é possível adicionar mapas nas aplicações baseados nos dados do Google Maps, como mostra a Figura 6.

A principal classe da API é o *MapView*, que automaticamente acede os servidores do Google Maps, baixa os dados, mostra os mapas e trata os eventos de toque no mapa.

Também é possível utilizar a API para adicionar marcadores, polígonos, camadas e alterar a visualização do utilizador de uma área específica.



Figura 6: Aplicação utilizando Google Maps API no Android

2.8.2 Hyper Text Markup Language (HTML)

HTML é a sigla de Hyper Text Markup Language (Linguagem de marcação de hipertexto), ou seja, a linguagem utilizada para criar páginas Web.

O HTML é a linguagem com que se escrevem as páginas web, podendo ser vistas pelo utilizador mediante um tipo de aplicação chamada navegador (browser). Podemos dizer portanto, que o HTML é a linguagem utilizada pelos navegadores, para mostrar as páginas web ao utilizador, sendo hoje em dia a interface mais extensa na rede. Podemos também acumular textos, imagens e áudios do nosso gosto.

2.8.3 Unified Modeling Language (UML)

Modelagem

A modelagem do sistema faz parte do processo de desenvolvimento de *software*, no contexto da Engenharia de Software. Utilizou-se na modelagem o padrão UML, por ser uma ferramenta que auxilia na modelagem de sistemas, do mais simples aos mais complexos, é mais adequado à orientação a objetos e proporciona um padrão para a preparação de planos de arquitetura de projeto, de sistemas, incluindo aspetos conceituais, como processos de negócios e funções de sistemas, além de itens concretos, como as classes escritas em determinada linguagem de programação, esquema de banco de dados e componentes de *softwares* reutilizáveis.

O Unified Modeling Language (UML) é uma linguagem de modelação *standard* aplicada em engenharia de software. O UML é utilizado para especificar, visualizar, modificar, construir e documentar os artefactos, durante a fase de desenvolvimento de um *software* orientado a objetos, para que o projeto satisfaça todos os requisitos. (ATHANAZIO 2010).

No contexto deste projeto, o UML vai ser utilizado para definir os modelos conceptuais de dados durante a fase de análise funcional de conteúdos e processos.

Diagramas UML

Os diagramas utilizados pela UML são compostos por nove tipos: diagrama de use case, de classes, de objeto, de estado, de sequência, de colaboração, de atividade, de componente e o de execução.

Todos os sistemas possuem uma estrutura estática e um comportamento dinâmico.

A UML suporta modelos estáticos (estrutura estática), dinâmicos (comportamento dinâmico) e funcionais. A modelagem estática é suportada pelo diagrama de classes e de objetos, que consiste nas classes e nos seus relacionamentos. Os relacionamentos podem ser de associações, herança (generalização), dependência ou refinamentos. Os modelos

dinâmicos são suportados pelos diagramas de estado, sequencia, colaboração e atividade. E a modelação funcional é suportada pelos diagramas de componentes e execução. (VIDEIRA, 2008).

2.8.4 Personal Home Page (PHP)

PHP é uma ferramenta abordada, que permite a criação de páginas dinâmicas, capaz de ser embebido dentro do código HTML e efetuar determinadas operações capazes de gerar páginas instantaneamente. Ainda o PHP é uma linguagem de programação recente para desenvolvimento de aplicações orientadas para a *World Wide Web*.

PHP é a combinação de uma linguagem de programação e de um servidor aplicacional. Este tipo de tecnologia está atualmente em crescente expansão, sendo um bom indicador do interesse que desperta perante a grande parte dos programadores web, na criação dinâmica de páginas. Funciona em, praticamente, todos os ambientes conhecidos sendo suportada por uma enorme comunidade na internet, que contribui para a sua difusão e aceitação ao nível mundial.

2.8.5 JavaScript

O *java Script* é uma linguagem utilizada na camada de comportamento e é uma linguagem de *scripting*, o que significa que tem de ser utilizada em conjunto com outra linguagem (como é o caso do HTML) ou aplicação.

A linguagem *JavaScript* é uma linguagem de *scripting* interpretada, que está disponível de forma gratuita na maioria dos *browsers* modernos.

JavaScript é uma linguagem com muitas possibilidades, permite a programação de pequenos scripts, mas também de programas maiores, orientados à objetos, com funções, estruturas de dados complexas, etc. Esta linguagem coloca à disposição do programador

todos os elementos que formam a página web, para que estas possa aceder a elas e modificá-las dinamicamente, se tornando o verdadeiro dono e controlador de cada coisa que ocorre na página. (CRIARWEB).

2.8.6 Base de Dados (BD)

Segundo Damas, (2005) uma base de dados (BD) consiste numa coleção de dados estruturados, organizados e armazenados de forma persistente.

Segundo (Carriço, 1996), um sistema de base de dados tem dois componentes fundamentais:

- A estrutura lógica e física, através da qual a informação é organizada;
- O sistema de gestão de bases de dados (SGBD), que assegura a gestão da informação.

O *software* que gere a informação contida numa base de dados é designado por sistema de gestão de base de dados (SGBD) e constitui a interface entre os dados e os utilizadores. O SGBD é responsável pela implementação do sistema, ao nível do *software*, “escondendo” do utilizador as complexidades da gestão interna dos dados e possibilitando-lhe uma visão lógica da estrutura da informação, compatível com o modelo conceptual. Um Sistema de Gestão de Base de Dados (SGBD) é um grupo de programas utilizados como interface entre uma BD e programas de aplicação ou uma BD e o utilizador.

O principal objetivo de um SGBD é prover um ambiente adequado e eficiente para armazenar e recuperar informações de uma base de dados. A gestão de dados envolve a definição de estruturas, para o armazenamento das informações e o fornecimento de ferramentas para manipulá-las.

Diversos SGBD de código livre existem, entre os quais: *MySQL*, *PostgreSQL* e Berkeley DB. Para este projeto, o SGBD utilizado foi o *MySQL*. Esta escolha se justifica quando

se leva em conta apenas uma característica do mesmo: gestão de utilizadores e permissões.

2.8.7 Phonegap

PhoneGap é um Framework livre e de código aberto que permite criar aplicativos móveis que utilizam web APIs padronizadas para as plataformas de que gostamos.

Também está voltado para o desenvolvimento de aplicações para dispositivos móveis, *open source*, multiplataforma, com suporte para os sistemas operacionais *iOS*, Google *Android*, *BlackBerry*, *Firefox OS*, *Ubutu*, *Windows Phone*, *Windows 8*.

Com ele é possível desenvolver aplicações utilizando as tecnologias *JavaScript*, *HTML5* e *CSS3*, dispensando assim o uso de *SDKs* ou compiladores, bastando apenas adicionar um arquivo *JavaScript* e um *JAR* para ter acesso a várias funções de *Smartphone*. O mesmo permite o acesso nativo aos recursos do *hardware* como GPS, Câmara, Acelerômetro, Geolocalização, Contactos, Sistemas de Arquivos, Notificações (alerta, som, vibração) e disponibilidade de rede.

Os limites do uso do *Phonegap* podem ser expandidos com o uso de *plugins*, permitindo ao desenvolvedor o acesso a outros recursos do aparelho. O Bluetooth Serial, por exemplo, permite que a aplicação desenvolvida se conecte e comunica com dispositivos utilizando o protocolo de comunicação sem fio Bluetooth. É suportado pelas plataformas *Android* e *iOS*, embora neste segundo seja necessário o uso de módulos do tipo BLE (Bluetooth Low Energy).

Já o *plugin SocialSharing* permite o compartilhamento de conteúdos, como textos, imagens, *links* e arquivos, com o auxílio do *widget* de compartilhamento nativo presente nas plataformas *Android*, *iOS*, e *Windows Phone*.

Além desses, é possível encontrar uma série de *plugins* desenvolvidos pela

comunidade que adicionam recursos que podem ser utilizados na criação de diversas aplicações.

No momento da compilação da aplicação, a API (Interface de Programação de Aplicativos) faz como aceder aos sensores, dados, status de rede, além de uma série de outros periféricos da plataforma alvo, na qual a aplicação irá ser executada, seguidos dos padrões de cada uma, simplificando assim a criação de uma mesma aplicação para plataformas diferentes.

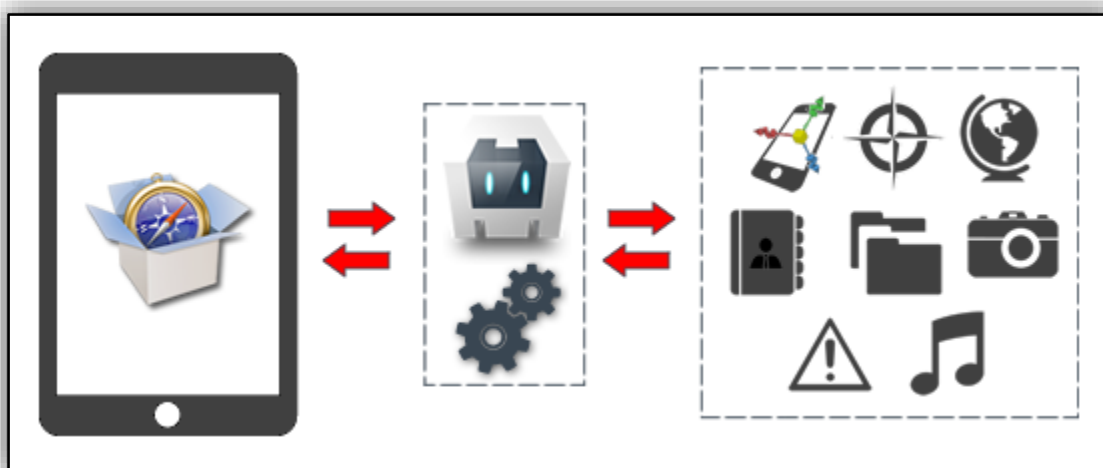


Figura 7: Estrutura do PhoneGap

Fonte: [Leonardo Moreira de Sousa]

A estrutura da Figura 7 ilustra a estrutura de funcionamento do *PhoneGap*. A interface com o utilizador é construída em HTML5 e CSS. A lógica da aplicação é implementada em *JavaScript*, que a API do *PhoneGap* interpreta e executa na plataforma nativa. Os acessos aos recursos de *hardware* são feitos através dos *plug-ins* desenvolvidos na linguagem nativa.

2.8.7.1 Arquitetura da Aplicação PhoneGap

A arquitetura típica de uma aplicação PhoneGap é que ela funciona como um cliente que interage diretamente com o utilizador, ao mesmo tempo em que ela se comunica com um servidor para envio e recebimento de dados.

O servidor da aplicação é, um servidor web que pode ser desenvolvido em diversos tipos de linguagem, desde que ele utilize os protocolos padronizados para a web. O servidor da aplicação também é o responsável por qualquer comunicação com servidores de dados.

A Figura 8 mostra como o PhoneGap atua somente na parte do cliente. Sua comunicação com o servidor ocorre por protocolos nos padrões web e o servidor é que tem a tarefa de comunicação com base de dados.



Figura 8:Arquitetura de uma Aplicação Phonegap

Fonte: [Leonardo Moreira de Sousa]

A arquitetura do lado do cliente procura manter o modelo *Single-page application*, ou seja, toda a lógica da aplicação está sob uma única página HTML que não sai da memória do aparelho e os dados são atualizados através do DOM, (Document Object

Model), uma API para HTML que permite com que programas e scripts pode aceder e atualizar o conteúdo e a estrutura de uma página (W3C, 2004).

Apesar de não ser muito comum, modelos *Multi-page application*, onde se utilizam diversas páginas para tratar a lógica da aplicação. O problema dessa solução é a perda de variáveis gravados na memória, o que aumenta a necessidade de comunicação com o lado do servidor, impactando assim na performance.

❖ Vantagens do *Phonegap*

A principal vantagem do *PhoneGap* é ser multiplataforma, uma vez que desenvolve a aplicação e com o mesmo código poderá ser utilizado para qualquer outra plataforma sem a necessidade de ser reescrito em java ou qualquer outra linguagem.

Uma outra vantagem do *Phonegap* é a possibilidade de compilar o mesmo projeto para as principais plataformas do mercado, como o *iOS*, *Android*, *Windows Phone*, *Blackberry*, *Bada*, etc.

2.8.7.2 Características do *Phonegap*

As APIs ficam disponíveis para uso através de *plug-ins* que provém às permissões, os acessos e os monitoramentos. Dependendo do sistema operacional, pode haver algumas particularidades que diferenciam o funcionamento do *plug-in*. Em seguida, veremos para que servem e como funcionam as principais funcionalidades desses recursos de *hardware*. O comportamento dessas APIs, quando a aplicação é executada num *browser* inova de acordo com a aplicação e com o browser, podendo gerar valores aleatórios.

O *phonegap* gera as seguintes funcionalidades:

❖ Acelerômetro

O acelerômetro é um dispositivo comum nos *smartphones* atuais e serve para medir forças de aceleração e alteração da sua orientação. Essas forças podem ser tanto estáticas, sendo afetadas pela gravidade, quanto dinâmicas, quando a aceleração é gerada através de movimentos manuais no aparelho.

❖ Câmera

Muitas aplicações requerem o uso da *câmera* disponível na aplicação, como os leitores de QR *Code* ou ainda código de barra bidimensional. Para atender essa demanda, o *PhoneGap* oferece o objeto *navigator.camera*, encontrado no Plug-in *org.apache.cordova.camera*, que oferece uma API para se tirar fotos e escolher imagens numa galeria.

A qualidade das fotos tiradas depende exclusivamente do *hardware* e todas as imagens exibidas pela aplicação serão apresentadas na melhor qualidade possível, mesmo que exista um parâmetro de qualidade que tente diminuir a qualidade da imagem.

Quando executado no Android, a aplicação *PhoneGap* pode parar de funcionar caso o aparelho tenha pouca memória. Isso acontece porque quando a aplicação nativo da câmera é solicitado, o Android inicia as atividades da câmera que podem consumir toda a memória disponível, fazendo com que o *PhoneGap* tenha sua execução interrompida.

❖ Contactos

Com o objeto *navigator.contacts* encontrado no Plug-in *org.apache.cordova.contacts*, é possível aceder a lista de contatos existentes. A lista de contatos é uma informação muito sensível, por questões de privacidade, por isso é aconselhado informar ao utilizador sempre que se for utilizar esse recurso.

No *Firefox OS* solicita a permissão ao utilizador, sendo necessário realizar a chamada das permissões, método no qual se solicita a permissão de leitura e/ou escrita. Já nos Sistemas

Windows, o objeto *contacts* apenas consegue ler as informações dos contatos, não sendo possível alterá-los.

❖ Arquivos

Para leitura e escrita de arquivos no dispositivo, o *PhoneGap* disponibiliza o Plug-in *org.apache.cordova.file* baseado na API de tratamento de arquivos do HTML5, o que facilita o trabalho dos desenvolvedores web. Nesse recurso, a reutilização de código é mais complicada, visto que o comportamento das pastas dos arquivos varia entre os sistemas operacionais.

❖ Geolocalização

Um dos recursos mais requisitados no desenvolvimento de aplicações, a geolocalização oferece os dados de localização do dispositivo, como latitude e longitude. Podem ser utilizadas as informações de GPS, dados de localização descobertos através de rede, como endereço de Ip, Wi-Fi, ou até mesmo de *bluetooth*. Informações de chip GSM também podem ser utilizadas.

❖ Base de Dados

O *PhoneGap*, por se tratar de desenvolvimento em *web view*, tem como principais opções APIs de armazenamento do lado do cliente. Na prática, isso significa que os dados passados na API são armazenados no próprio dispositivo, no mesmo lugar onde se salvam as preferências do utilizador. Através da API, também é possível recuperar, manipular e realizar a pesquisa dos dados.

Todas as APIs possuem em comum o fato de limitarem a aceder informações armazenadas ao mesmo domínio e porta que salvou os dados. Elas também sofrem restrições de tamanho permitido por domínio, que variam de acordo com o sistema.

2.8.8 Eclipse

O Eclipse é um ambiente de desenvolvimento integrado (IDE- Integrated Development Environment), vulgarmente utilizado para desenvolver e testar aplicações em linguagens tais como java, C, C++, etc. É também o IDE recomendado para o desenvolvimento de aplicações Android. A mais recente versão é o Eclipse (4.2).

2.8.8.1 Principais Características do Eclipse

O Eclipse é um IDE de código aberto utilizado para desenvolvimento da linguagem java, mas é também multilinguagem, ou seja, suporta outras linguagens de programação tais como C/C++ instalando-se os devidos plug-ins adicionais para cada linguagem. É um projeto livre de patentes, por ser um software livre, e sua aplicação funciona em vários ambientes.

A plataforma do Eclipse fornece vários pacotes de desenvolvimento, tais como Eclipse JDT, a base para qualquer plug-in na linguagem java, o Eclipse SDK, que é o pacote de distribuição da IDE java, o Eclipse WTP (*Web Tools Platform*), utilizado para desenvolvimento de linguagem para web, e o compilador do JDT, que é seu próprio compilador java, que é mais rápido e de código aberto.

O Eclipse permite a refatoração do código, que é uma forma organizada de reestruturar o código para ser melhorado. Um aspeto importante de uma refatoração é que ela melhora o design sem mudar a semântica do design, não adicionando nem removendo sua funcionalidade. Alguns exemplos de refatoração de código é renomear métodos, encapsular campos, extrair classes, introduzir afirmações e especializar os métodos.

Todo sistema operacional contém vários componentes que compõem sua interface de utilizador. Estes componentes incluem botões, janelas, menus e tudo o que se vê na tela de computador.

3 ANÁLISE DO SISTEMA

3.1 Caracterização da Plataforma Roteiro Turístico Virtual em São Vicente

A cidade do Mindelo localiza-se na ilha de São Vicente e fica situada no grupo das ilhas do Barlavento no noroeste do arquipélago. É a segunda cidade mais populosa de Cabo Verde com cerca de 76.140 mil habitantes, apesar de ser uma das ilhas mais pequenas do país. São Vicente é uma pequena ilha vulcânica, com apenas 227 km², com um comprimento máximo de 24 km e uma largura de 16 km. Foi descoberta a 22 de Janeiro de 1462 e até 1838 esteve praticamente despovoada. Apenas em meados do século XIX, começou a ser povoada, depois dos ingleses terem instalado no Mindelo um depósito de carvão para reabastecer os navios que percorriam o oceano Atlântico.

Mindelo tornou-se assim, através do Porto Grande, uma cidade universal onde encontravam-se navios de diversas origens, para o reabastecimento de carvão depois de uma longa viagem. O desenvolvimento do porto natural de São Vicente, protegido com condições propícias para a embarcação e de qualquer tipo de navio, teve um importante contributo para o progresso da economia do arquipélago. Pois, com a atividade carvoeira estabelecida e bem-sucedida no porto da ilha, o estado conseguiu obter rendimentos elevados, que lhe permitiram o desenvolvimento das outras ilhas vizinhas, tornando-o o principal porto de Cabo Verde, por onde passaram grande parte das importações do país.

Durante alguns anos o porto do Mindelo perdeu alguma da sua importância comercial local e nacional.

O Porto Grande serve de escala transatlântica para navios de todo mundo. Mindelo é a capital da diversão, com restaurantes, Hoteis, Discotecas, Praias, Pensões, Farmácias, frequentes animações com música ao vivo, o carnaval mais marcante de Cabo Verde.

Tem um património histórico bem preservado e um movimento cultural permanente. Diz-se que é a cidade mais acolhedora e atraente do país.

Na cidade do Mindelo existem principais pontos turístico, referenciados no parágrafo anterior entre outros de forma simples, clara e objetiva para os turistas.

A análise do sistema é vista como uma atividade onde se pensam e descrevem os sistemas de informações, que assume de forma decisiva a importância no processo de desenvolvimento de sistemas.

Esta análise encontra-se dividida em três secções, em que a primeira tem como objetivo a definição dos requisitos necessários para o funcionamento do sistema, na segunda são descritos os casos de usos, que permitem a interação dos utilizadores com o sistema e a terceira corresponde a modelação de dados através dos diagramas de caso de uso, classe, sequência e entidade relacionamento.

3.2 Levantamento de Requisitos

Os requisitos será pré-requisito de grande importância para o sucesso do projeto e a produção de um *software* de qualidade. Todos os projetos, independentes do seu tamanho, podem beneficiar da atenção dada aos requisitos. Os requisitos definem os serviços que o sistema deve oferecer e as restrições aplicáveis à sua operação (CARVALHO; TAVARES, 2002).

Os requisitos são divididos em, funcionais e não funcionais:

- ❖ Os **Requisitos Funcionais** - são requisitos que expressam funções ou serviços que um *software* deve ou pode ser capaz de executar ou fornecer. As funções ou serviços são geralmente processos que utilizam entradas para produzir saídas.
- ❖ Os **Requisitos Não Funcionais** - são requisitos que declaram restrições ou atributos de qualidade no processo de desenvolvimento de *software*.

3.2.1 Requisitos Funcionais (RF) dos aplicativos

Código	Descrição
RF01	O sistema deverá possuir entrada de dados para autenticação através de utilizador e password;
RF02	Deve mostrar o mapa da cidade do Minho com as ruas e o nome das ruas;
RF03	O sistema deverá possuir três camadas no mapa, a camada que mostra os pontos turísticos, camada que mostra o percurso da linha turismo e a camada de pontos de parada da linha turístico;
RF04	O utilizador poderá habilitar e desabilitar cada uma das camadas no mapa;
RF05	O turista poderá buscar rapidamente um ponto turístico através das coordenadas por ordem alfabética com os nomes dos pontos turísticos;
RF06	O turista poderá obter informações detalhadas ao clicar sobre o ícone do ponto turístico;
RF07	A aplicação deverá permitir o controlo do mapa, como mover, aproximar e afastar o mapa com movimentos de toque no ecrã;

Tabela 3: Requisitos Funcionais da aplicação

3.2.2 Requisitos Não Funcionais (RNF)

Código	Descrição
RNF01	A aplicação deverá ser simples e intuitiva para o utilizador;
RNF02	A aplicação deverá responder rapidamente aos comandos no ecrã;
RNF03	O sistema deverá informar o utilizador que deve aguardar, através de ícone animado ou mensagem no ecrã, durante operações demoradas;
RNF04	A aplicação deverá tratar erros inerentes ao sistema, como falha na conexão de dados e na localização geográfica do dispositivo;
RNF05	A aplicação deve ser funcional em <i>smartphones</i> e <i>tablets</i> ;
RNF06	O aplicativo será executado em sistema operacional Android.

Tabela 4: Requisitos não Funcionais

3.3 Lista de Caso de Uso

Os casos de uso descrevem as funcionalidades específicas que um sistema deve executar. Cada caso de uso descreve um possível cenário de interação que um sistema externo ou outra entidade tem com o sistema a ser desenvolvido.

Na tabela 5 estão identificados os casos de uso que foram necessários para o sistema.

Caso de Uso	Descrição
Visualizar Pontos Turísticos	Mostra informações detalhadas do ponto turístico.
Buscar Pontos Turísticos	Se clicar em Buscar Pontos Turísticos mostra a lista dos pontos turísticos em ordem alfabética para localização rápida no mapa.
Mostrar Mapa	Mostra detalhes do mapa de São Vicente com as camadas selecionadas.
Listar Pontos Turísticos	Listar todos os pontos turísticos.
Login do Sistema	O Utilizador faz login no sistema para configurar e atualizar o mapa.
Configurar Mapa	Configura o mapa e atualiza os pontos turísticos.
Selecionar Camada do Mapa	Mostra lista de seleção das camadas do mapa: pontos turísticos, itinerário da linha de turismo e pontos de paradas da linha de turismo.

Tabela 5: Lista de Caso de Uso

3.4 Modelação de dados

A modelação de dados permite a aplicação de modelos que asseguram correta definição de funcionalidades.

3.4.1 Diagrama caso de uso

O diagrama de caso de uso permite visualizar e especificar o comportamento do administrador e do turista com o sistema.

Todos os atores que afetam os casos de usos foram definidos seguindo um perfil de acessos e permissões ao sistema:

- **Administrador:** é o ator que possui acesso a todos os recursos do sistema;
- **Turista:** é o ator com o perfil mais completo. Tem acesso a todas as informações técnicas do mapa.

A figura 9 representa o diagrama de caso de uso que demonstra a forma com que o ator interage com o sistema.

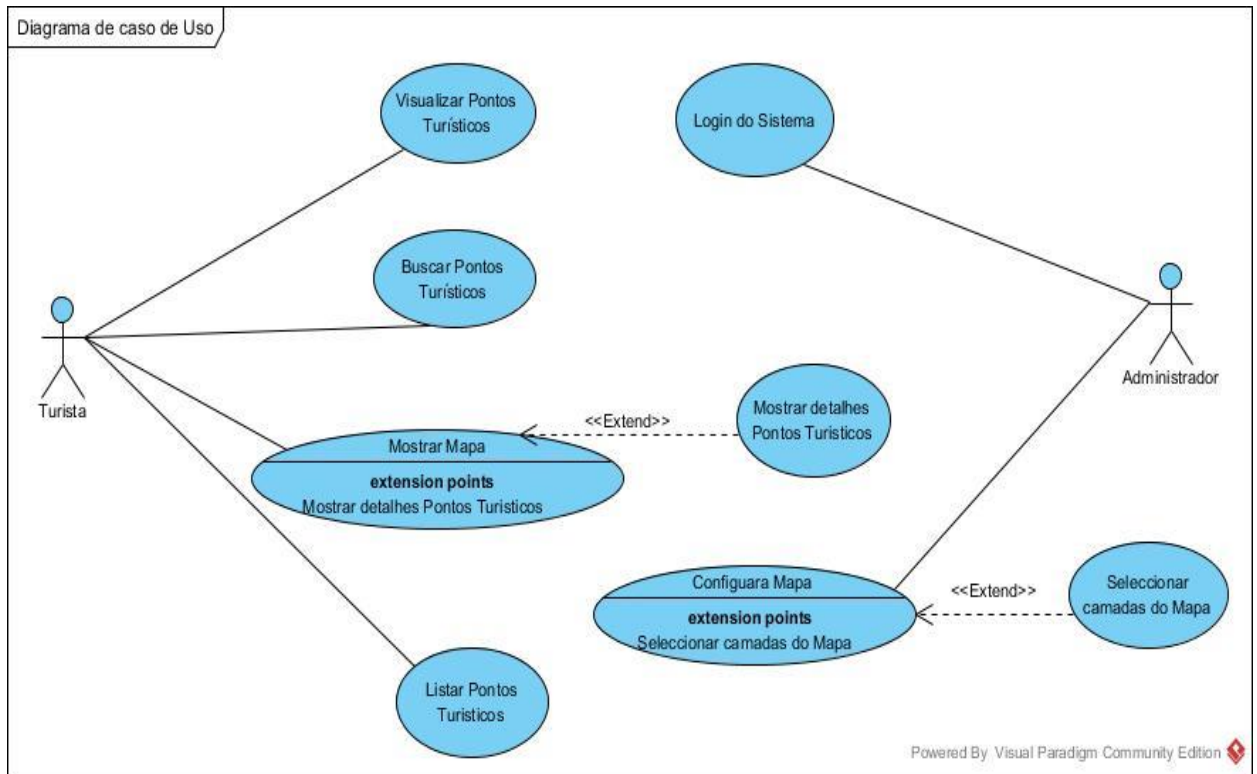


Figura 9:Diagrama de Caso de Uso

3.4.2 Diagrama de Classes

O diagrama de classe representa a classe envolvida no sistema e os relacionamentos existente entre elas, como mostra a figura 10.

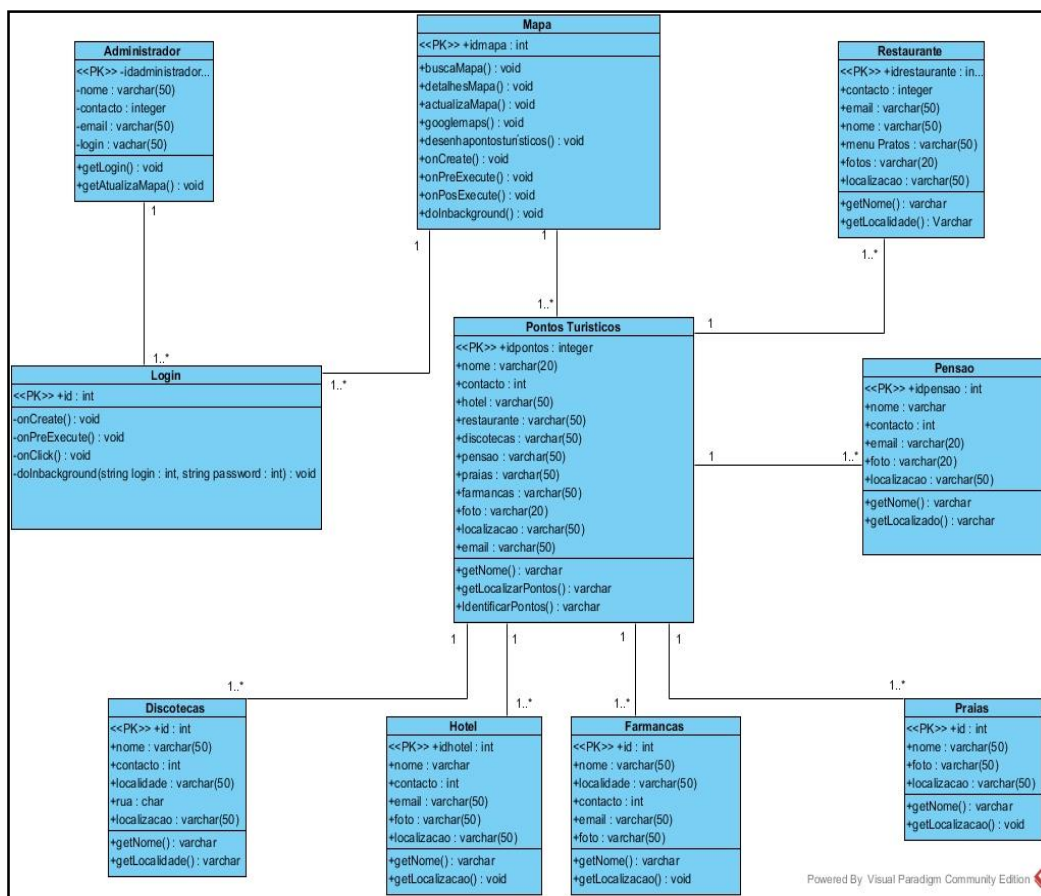


Figura 10:Diagrama de Classe

3.4.3 Diagrama de sequência

Os diagramas de seqüências representam as interações dos autores (administrador e turista) o envio de mensagens de solicitação e respostas de retorno entre eles, em uma linha de tempo.

A figura 11 representa o diagrama de sequência para aceder ao sistema, onde o Turista visualiza os pontos turísticos. Após aceder ao sistema, o turista poderá escolher qualquer ponto a sua preferência.

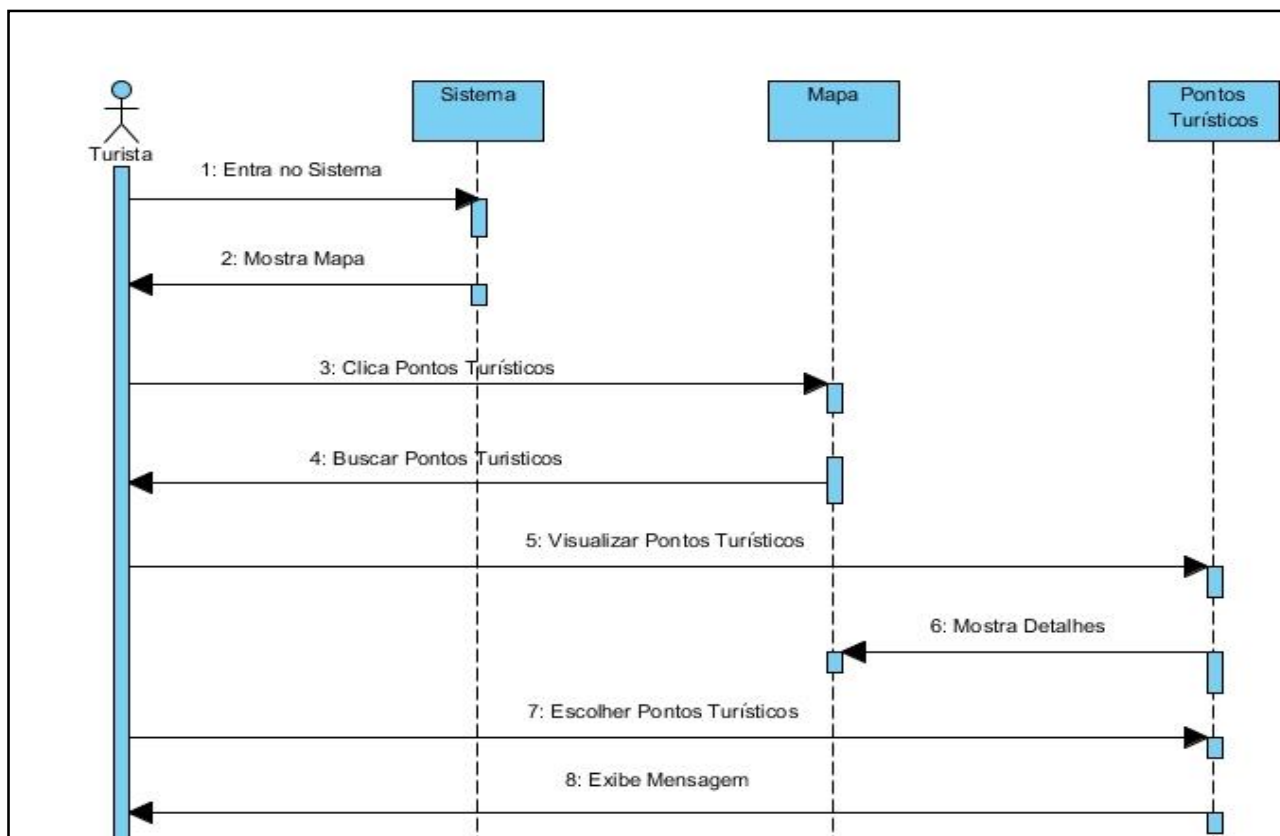


Figura 11:Diagrama de Sequência de Acesso ao Sistema

A figura 12 representa o diagrama de login do utilizador, onde pode fazer login no sistema para atualizar e configurar o mapa.

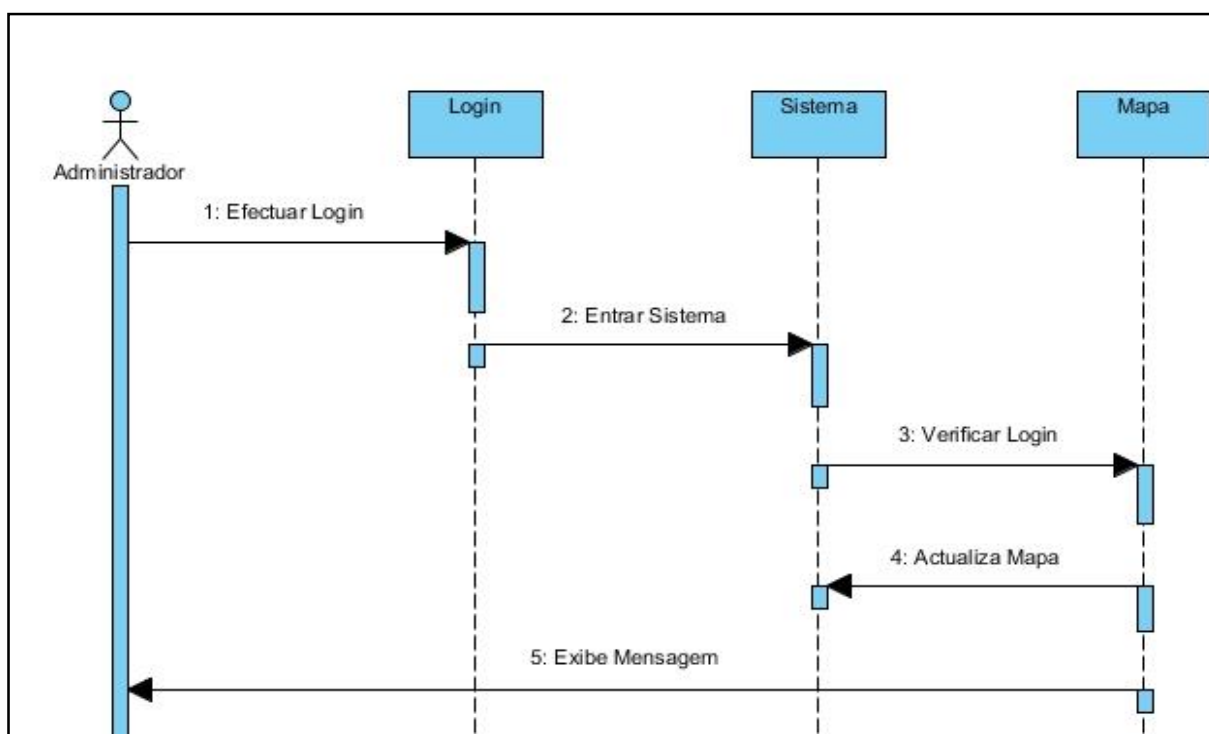


Figura 12: Diagrama de Sequência de Login no Sistema

A figura 13 representa o diagrama de entidade e relacionamento que relaciona e expressam as dependências e exigências entre o sistema.

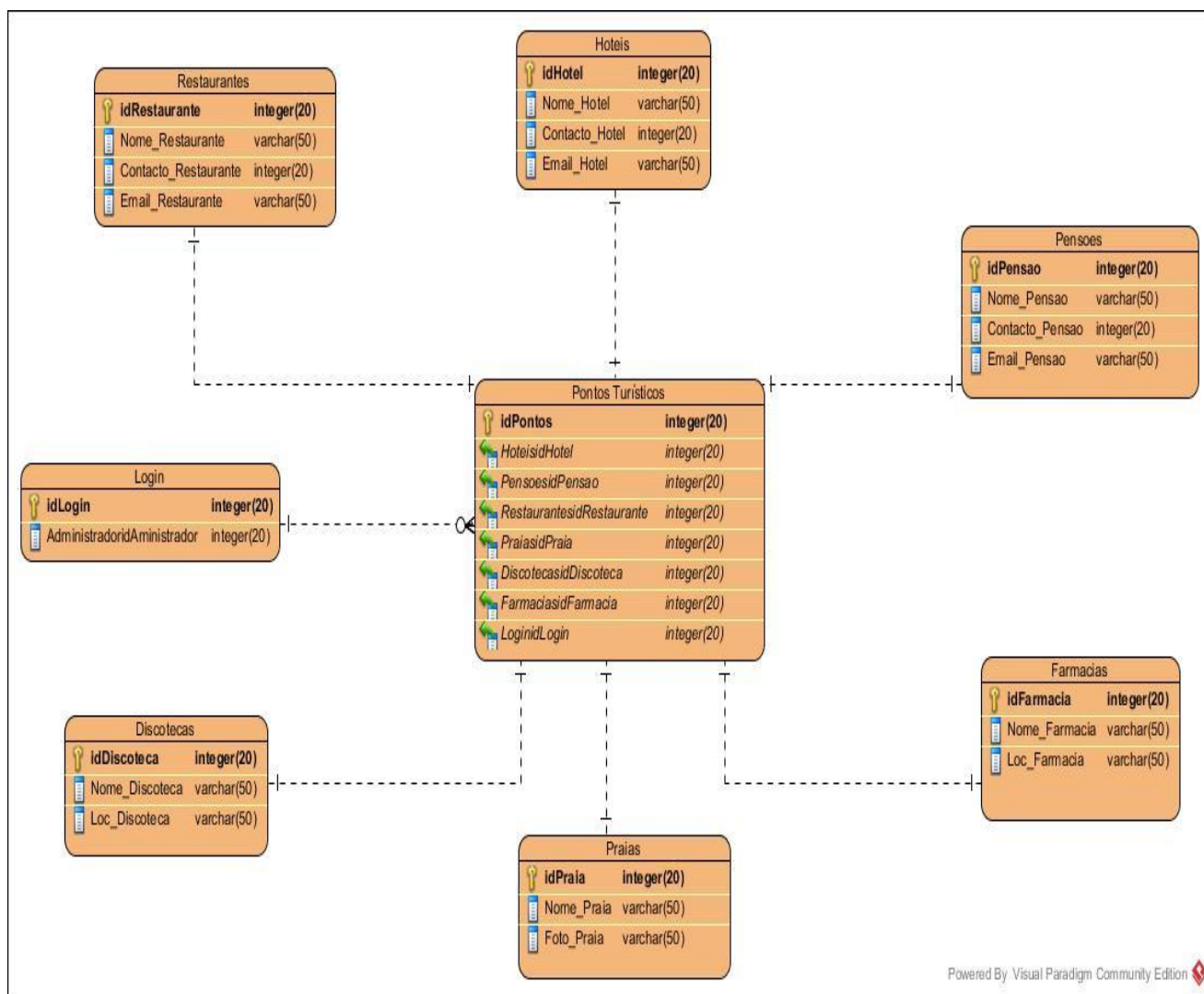


Figura 13: Diagrama de Entidade e Relacionamento ao Sistema.

4 PROTÓTIPO DO SISTEMA

Este capítulo tem como objetivo abordar as principais características do protótipo do sistema proposto no presente trabalho.

4.1 A Plataforma Roteiro Turístico Virtual

A aplicação Turístico é uma aplicação para dispositivos com o sistema operativo Android, capaz de disponibilizar informações referentes aos pontos turísticos como Farmácias, Hoteis, Restaurantes, Praias, Discotecas e Pensões do Minho. A plataforma turística, também mostra todas as informações detalhadas dos pontos para os turistas. As interfaces de utilizador foram projetadas com o uso da linguagem PHP e no código fonte foram utilizadas as linguagens de programação HTML e JavaScript.

4.2 Estrutura do projeto PR Turístico Virtual

Depois de ter terminado o funcionamento da aplicação Android, agora é a explicação como se encontra a estrutura do projeto Android na seguinte figura.

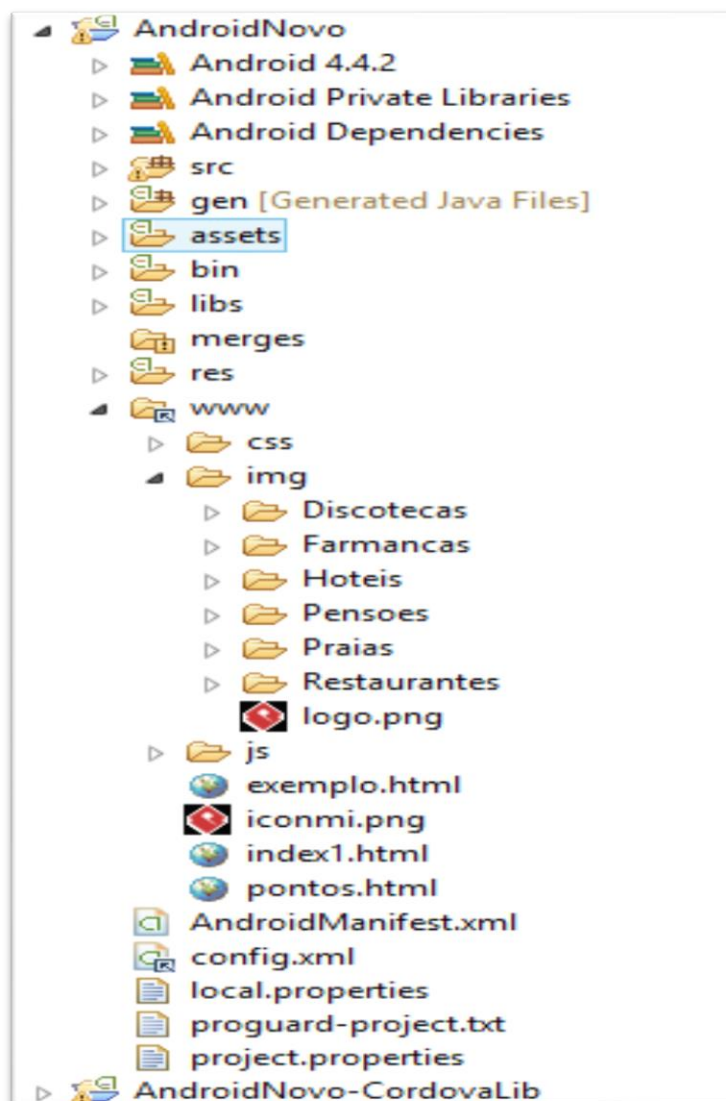


Figura 14: Estrutura do Projeto Plataforma Roteiro Turístico Virtual

A aplicação Plataforma Android possui os seguintes componentes:

- ❖ src: A pasta src armazena a Activity principal e todas as outras classes que compõem o código fonte Java de uma aplicação em Android. A figura 14 mostra

a Activity principal denominada de MainActivity e os demais ficheiros que compõem o código fonte HTML da aplicação Plataforma Turístico;

- ❖ img: Onde fica armazenada as imagens dos pontos turísticos inseridos no sistema (discotecas, farmácias, restaurantes, etc).
- ❖ js: É a pasta dos ficheiros que compõem o código fonte em HTML do mapa para ser visualizado pelos turistas.

4.3 Desenvolvimento de API Google Maps

O API Google Maps é um serviço gratuito, desenvolvido pelo Google, que permite a pesquisa e a visualização de mapas de ruas, imagens de satélite da terra, bem como utilizar as funções para o desenho de rotas.

Para utilizar o API Google Maps é necessário estar registado com o serviço do Google Maps, concordar com os termos de serviço e obter uma chave de autenticação. O API Google Maps é utilizado para possibilitar aos utilizadores a visualização de locais e rotas de destino. Para ter acesso ao API do Google Maps é necessário conhecer o seguinte código que se apresentam na figura.

```
<title>Eg. Google Maps Marker using External JSON</title>
<style type="text/css">
html { height: 100% }
body { height: 100%; margin: 0; padding: 0 }
#map_canvas { height: 100% }
</style>
<link rel="stylesheet" type="text/css" href="css/style.css"/>
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.min.js"></script>

<script type="text/javascript" src = "http://maps.google.com/maps/api/js?sensor=false">

</script>
<script type="text/javascript">
/* function addInfoWindow(marker, message) {
```

Figura 15:Código do mapa em API Google Maps

A figura 16 representa o código com a função onde inicializa o mapa, a variável infowindows que exibe textos ou imagens em uma janela pop-up acima do mapa, o center é onde está centralizado com as coordenadas (latitude e longitude), o tamanho que é estabelecido e ainda o map_canvas é onde está localizado o mapa.

```
function initialize() {
var infowindows = [];
var mapOptions = {
center: new google.maps.LatLng(16.8765460,-24.9813500),
zoom: 10,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
prenchermapa();
}
```

Figura 16:Localização do Mapa

4.4 Descrição das Funcionalidades do Sistema

O sistema apresenta vários tipos de pontos turísticos que serão demonstrados a seguir.

4.4.1 Autenticação

São potenciais utilizadores do sistema, todos os turistas que interagem com a aplicação, no sentido de usufruir das funcionalidades dos pontos turísticos.

4.4.2 Menu Principal

O menu principal estará sempre disponível, variando as várias opções de acordo com o utilizador. Estão disponíveis as seguintes opções de menu, que se desdobram nos seguintes itens.

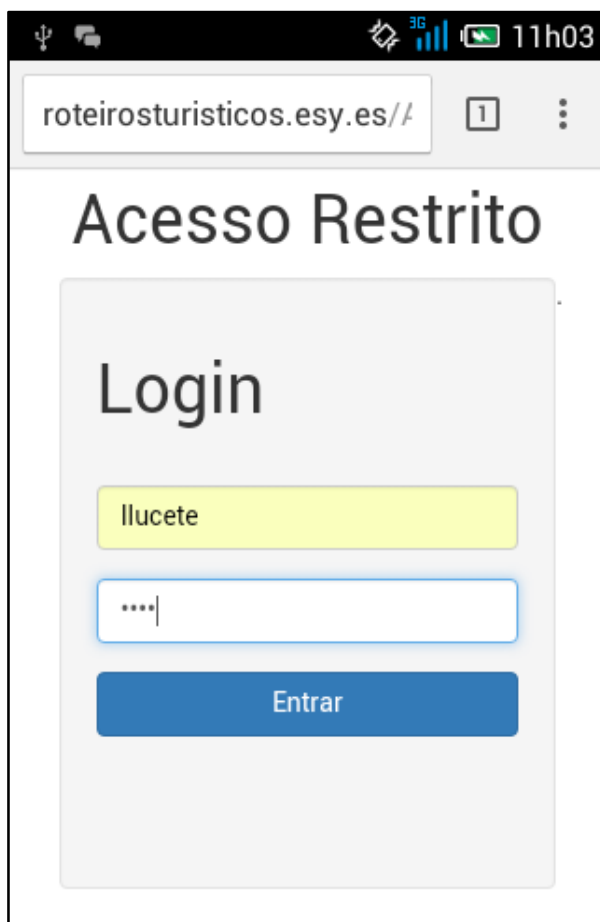
Pontos Turísticos:

- ❖ Restaurantes;
- ❖ Hoteis;
- ❖ Pensões;
- ❖ Discotecas;
- ❖ Farmácias;
- ❖ Praias.

4.4.3 Interfaces Principais do Sistema

As interfaces do sistema permitem aceder e visualizar informações dos pontos turísticos. A seguir serão apresentadas as interfaces do sistema com a descrição dos seus objetivos e funcionalidades.

A figura 17 apresenta a interface do login do sistema, que é a página inicial, que permite o acesso a entrada de utilizadores, que são cadastrados pelo administrador do sistema. Se o utilizador estiver cadastrado no sistema terá acesso a página da gestão dos pontos. Se o login e senha forem válidos, o sistema habilita o menu inicial. Para os utilizadores ainda não cadastrados no sistema, só têm acesso a página do mapa como os turistas.



roteirosturisticos.esy.es/

Acesso Restrito

Login

Ilucete

....

Entrar

Figura 17: Interface de Login

A figura 18 é a interface responsável pelos Pontos Turísticos ou Gestão de Pontos. É nessa interface que insere os pontos turísticos com todas as informações básicas. O mapa aqui presente serve para localizar os pontos através da longitude e latitude.



roteirosturísticos.esy.es/ 1

Pontos

Nome
N Baia das gatas

Localidade
M Mindelo

Endereço
A Baia dad gatas

Telefone
T 2315623

Email
@ baia@gmail.com

Figura 18: Interface de Cadastro de Pontos Turísticos



Figura 19:Interface do Mapa do Ponto Cadastrado

A figura 20 é a interface responsável pela gestão de utilizadores, onde posso cadastrar os utilizadores para ter acesso ao sistema.



The screenshot shows a mobile application interface for user registration. At the top, the status bar displays icons for signal, battery, and the time 13h08. Below the status bar is a browser address bar showing the URL 'roteirosturisticos.esy.es/'. The main title of the form is 'Utilizadores'. The form consists of several input fields, each with a label and a corresponding icon: 'Nome' (Name) with a person icon, 'Endereço' (Address) with a house icon, 'Senha' (Password) with a key icon, 'Repita a Senha' (Repeat Password) with a key icon, and 'Telefone' (Phone) with a phone icon. The values entered in the fields are 'Ilucete', 'Ribeirinha', four dots for the password, four dots for the repeat password, and '9836629' for the phone number.

Nome
Ilucete

Endereço
Ribeirinha

Senha
....

Repita a Senha
....

Telefone
9836629

Figura 20:Interface de Cadastro dos Utilizadores

A figura 21 representa a aplicação com a interface principal do sistema, com o mapa onde estão inseridas todas as informações acessíveis para os turistas. O turista pode visualizar e escolher todos os pontos turísticos, como: Farmácias, Hoteis, Restaurantes, Pensões, Praias, Discotecas.

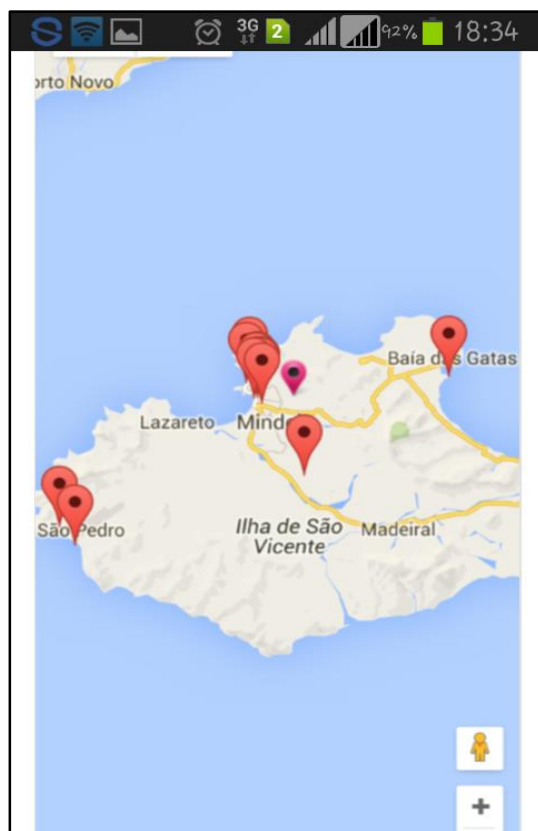


Figura 21:Interface do Mapa

A figura 22 representa a interface do mapa com os pontos de interesse dos turistas e o mapa representa somente os pontos selecionados, e terá escondido os outros para serem visualizados quando for necessário.

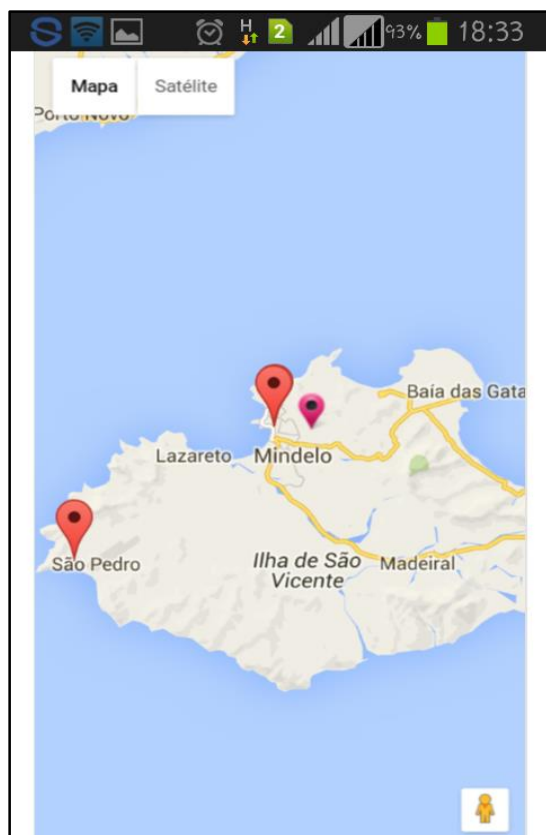


Figura 22:Interface do Mapa Pontos de Interesses

As interfaces seguintes representam as informações sobre os determinados pontos. Com essas informações adicionadas no mapa o turista pode localizar o ponto com mais simplicidade.

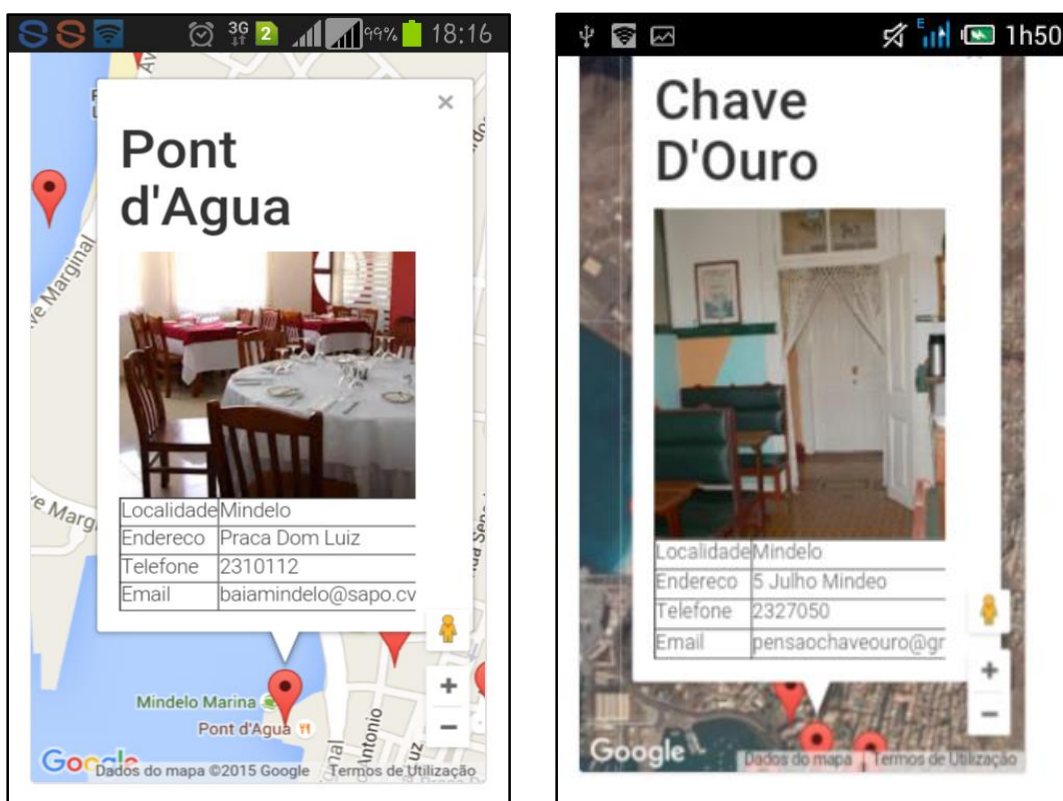


Figura 23: Interfaces com informações dos Pontos

5 CONCLUSÃO

Durante a realização do trabalho foi projetada e desenvolvida uma aplicação para dispositivos móveis com o sistema Android, através do uso das tecnologias, principalmente os serviços de localização como (GPS), Rede, Google Maps API e das tecnologias utilizadas (PHP, HTML, Mysql, Javascript).

Com os objetivos específicos neste trabalho nomeadamente informar aos turistas os pontos de acessos mais importantes como os Hoteis, Restaurante, Praias, Discotecas, Farmácias, Pensões, efetuar pesquisas de lugares (finalidade, tipo de lugar, localização, etc.), cria um sistema que possibilita a expansão do aplicativo para outras plataformas, também móveis, e que permita uma integração com outros sistemas turísticos.

Também foi abordada a importância do turismo e as dificuldades que um turista possa ter num lugar desconhecido. Com isso em mente foi traçado como objetivo principal do trabalho criar um protótipo com pontos roteiro turísticos, de interesse e ajuda-lo na navegação.

Ao final do trabalho, como resultado da implementação, foi obtido um protótipo de sistema Android para gerir roteiros turísticos com uma base de dados armazenada através da plataforma Google App Engine. Existe ainda condições para melhorias e novas funcionalidades, porém foi demonstrado como este protótipo utilizando LBS e plataformas móveis pode facilitar a vida dos turistas.

5.1 Dificuldades na implementação do sistema

As dificuldades encontradas na implementação do sistema tem como objetivo dar a conhecer os problemas, para ajudar a desenvolver formas de os ultrapassar.

Um das dificuldades deste projeto resultam de pouca bibliografia encontrada sobre a plataforma roteiro turístico virtual, que permitia abordar os componentes teóricos deste trabalho.

5.2 Trabalhos Futuros

Como perspetiva de atividades futuras, dando continuidade ao processo de análise de resultados deste trabalho, pretende-se e adicionar novas funcionalidades ao sistema, como, por exemplo:

- Criar mais pontos turísticos no mapa principalmente para as outras ilhas do País;
- Colocar todas as informações detalhadas do mapa, em diversos idiomas, para os turistas escolherem;
- Pretende-se otimizar o mapa de modo a disponibilizar mais que uma rota e também desenvolver a aplicação para dispositivos como o sistema IOS, e BLACK BERRY.

Com o desenvolvimento da plataforma roteiro turístico virtual, apesar de todas as dificuldades e barreiras, pertencentes a um projeto de *software*, consegui atingir o objetivo. Esta experiência proporcionou uma grande satisfação e conhecimento da plataforma Android.

REFERÊNCIAS BIBLIOGRÁFICAS

Silva, Alberto. (2001) *Metodologia e Ferramentas Case, Volume 1, 2º Edição*. Editora de Informática.

Silva, Alberto. (2001) *Metodologia e Ferramentas Case (Linguagem de Modelação UML, Metodologia e Ferramentas CASE na Conceção e Desenvolvimento de Software)*. Edições Centro Atlântico, Portugal.

Serrão, Carlos. (2007) *Programação com PHP 4.3, 2º Edição*. Editora de Informática.

Videira, Carlos. (2001) *Metodologia e Ferramentas Case, Volume 1, 2º Edição*. Editora de Informática.

SILVA, Alberto e VIDEIRA, Carlos (2008). *UML Metodologia e Ferramentas CASE*. 2ª Edição. Editora Centro Atlântico.

Amaral Luiz, Varajão João, (2006). *Planeamento de sistemas de informação, 3ª Edição*. Editora de Informática.

Melo, João. (1985) *União dos Escritores Angolanos, volume 36 de Cadernos Lavra & oficina*. Edita de Informática.

Queirós, Ricardo. (2013). *Android - Introdução ao desenvolvimento de Aplicações*. Editora de Informática.

S. Pressman, Roger. (2005) *Engenharia de Software 5º Edição*. Editora de Informática.

Pereira, José (1998). *Tecnologia de Base de Dados. 3ª Edição*. FCA Editora de Informática.

Lechetta, Ricardo (2010). *Aprenda a criar aplicações para dispositivos Móveis 3º Edição*.

Damas, Luís (2005). *SQL – Structured Query Language”, FCA*.

INTERNET:

Pesquisa de Introdução. Disponível em

<https://www.lume.ufrgs.br/bitstream/handle/10183/66088/000870837.pdf?sequence=>

Consultado em [05/01/2015].

Localização baseados em serviços. Disponível em [http://www-di.inf.puc-](http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/04/AdolfoCorreia-Mono.pdf)

[rio.br/~endler/courses/Mobile/Monografias/04/AdolfoCorreia-Mono.pdf](http://www-di.inf.puc-rio.br/~endler/courses/Mobile/Monografias/04/AdolfoCorreia-Mono.pdf). Consultado em [08/01/2015].

IDC. Sistema Operativo para Dispositivos Móveis. Disponível em:

<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>, Consultado em: [10/01/2015].

Documentação oficial do Android. Disponível em

<http://www.developer.android.com/guide/index.html>. Consultado em [05/02/2015].

Arquitetura de Android: Disponível em

<http://www.softblue.com.br/blog/home/postid/11/CONHECENDO+O+ANDROID>
Consultado em [02/05/2015].

Arquitetura do Google Android: Disponível em <http://developer.android.com>. Consultado em [13/06/2014].

Sistema Operacionais para dispositivos móveis: Disponível em

<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Consultado em [02/08/2015].

Componentes de uma aplicação Android: Disponível em

<http://www.developer.android.com/guide/index.html>. Consultado em [20/08/2014].

Ciclo de vida de uma Activity: Disponível em:

<http://developer.android.com/guide/components/activities.html>. Consultado em [20/08/2014].

Henrique. F (2008). Material de Ensino/Introdução ao Eclipse. Disponível em <http://www.cin.ufpe.br/~phmb/ip/.html>. Consultado em [03/09/2015].

Linguagem de Modelação Unificada. Disponível em

ATHANAZIO.http://www.athanazio.com/downloads/analise/apostila_uml.pdf. Consultado em [24/09/2015].

Documentação de PHP. Disponível em:

http://www.cin.ufpe.br/~ags/2464_php_com_mysql.pdf. Consultado em 05/10/2015.

Documentação oficial do PhoneGap. Disponível em

<http://www.phonegap.com>. Consultado em [09/10/2015].

Introdução ao Android Studio. Disponível em

<<http://developer.android.com/sdk/index.html>>. Consultado em [15/09/2015].

Google Maps JavaScript API v3. Disponível em

<<https://developers.google.com/maps/documentation/javascript/>>. Consultado em [25/09/2015].

Google Maps JavaScript API v3. Disponível em <<http://mygeoposition.com/>>.

Consultado em [26/09/2015].